



MySQL-MGR APIs

[MySQL MGR](#)[MGR Backup](#)[MGR Resto](#)[MGR Schedule](#)[MGR User](#)



MySQL MGR

Description

MySQL is the Schema for the mysqls API

Type

object

Specification

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources

Property	Type	Description
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	MysqlSpec defines the desired state of Mysql
status	object	MysqlStatus defines the observed state of Mysql

.spec

Description

MysqlSpec defines the desired state of Mysql

Type

object

Required

version

Property	Type	Description
crVersion	string	Deprecated: use upgradeOption.CrVersion
ipFamilyPrefer	string	IPFamilyPrefer is the preferred IP family
mgr	object	MGR use to configure MySQLCluster
params	object	Params is the configuration of MySQL Server
paras	object	Paras deprecated: this field is deprecated, please use spec.params instead
password	string	Password Deprecated: this field is deprecated
pause	boolean	Pause is the flag to pause the MySQL cluster
pxc	object	PXC use to configure PerconaXtraDBCluster
runAsRoot	boolean	RunAsRoot is the flag to run MySQL Server as root
s3Config	object	S3Config is the configuration of S3 for PXC backup

Property	Type	Description
<code>upgradeOption</code>	<code>object</code>	UpgradeOption is the option to upgrade the MySQL cluster
<code>useSafeConf</code>	<code>boolean</code>	UseSafeConf is the flag to use safe configuration
<code>version</code>	<code>string</code>	Version is the version of the MySQL Server

.spec.mgr

Description

MGR use to configure MySQLCluster

Type

`object`

Property	Type	Description
<code>affinity</code>	<code>object</code>	If specified, affinity will define the pod's scheduling constraints
<code>backupVolumeClaimTemplate</code>	<code>object</code>	BackupVolumeClaimTemplate allows a user to specify a volume to temporarily store the data for a backup before it being shipped to object storage.
<code>baseServerId</code>	<code>integer</code>	BaseServerID defines the base number used to calculate a unique server_id for MySQL instances in the cluster

Property	Type	Description
		Valid range 1 to 4294967286. If omitted in the manifest file (or set to 0) defaultBaseServerID value will be
<code>certConfig</code>	<code>object</code>	CertConfig allows a user to specify custom CA certificate with advanced option.
<code>config</code>	<code>object</code>	Config allows a user to specify a custom configuration file for MySQL.
<code>configuration</code>	<code>string</code>	Mysql configuration
<code>enableStorage</code>	<code>boolean</code>	EnableStorage allow a user to enable mysql storage
<code>image</code>	<code>object</code>	Image allows a user to specify image.
<code>imagePullPolicy</code>	<code>string</code>	PullPolicy describes a policy for if/when to pull a container image
<code>imagePullSecret</code>	<code>array</code>	ImagePullSecret defines the name of the secret that contains the required credentials for pulling from specified Repository.
<code>ipFamilyPolicy</code>	<code>string</code>	IpFamilyPolicy means the IP family policy for cluster

Property	Type	Description
<code>ipFamilyPrefer</code>	<code>string</code>	IpFamilyPrefer means the preferred IP family for
<code>jemallocConf</code>	<code>string</code>	JemallocConf means the jemalloc configuration
<code>majorViewIDQuorum</code>	<code>boolean</code>	MajorViewIDQuorum means use the major view ID to determine the quorum
<code>members</code>	<code>integer</code>	Members defines the number of MySQL instance cluster. InnoDB clustering supports between 1-9 members.
<code>monitor</code>	<code>object</code>	Monitor define MySQL monitor spec
<code>multiClusterReplication</code>	<code>object</code>	Config this field to support multi cluster replication disaster recovery.
<code>multiMaster</code>	<code>boolean</code>	MultiMaster defines the mode of the MySQL cluster. If set to true, all instances will be R/W. If false (the default), only a single instance will be R/W and the rest will be R/O.
<code>nodeSelector</code>	<code>object</code>	NodeSelector is a selector which must be true for a pod to fit on a node. Selector which must match a node's labels for the pod to be scheduled on that node. For more info: https://kubernetes.io/docs/concepts/configuration/assign-pod-node/

Property	Type	Description
		https://kubernetes.io/docs/concepts/configuration/pod-node/ ↗
<code>paras</code>	<code>object</code>	Paras provide paras template feature
<code>patches</code>	<code>object</code>	Patches use to apply patches to the MySQL cont
<code>pause</code>	<code>boolean</code>	Pause means pause the cluster
<code>podCIDR</code>	<code>string</code>	PodCIDR allows a user to specify custom CIDR.
<code>podManagementPolicy</code>	<code>string</code>	PodManagementPolicy defines the policy for crea pods under the statefulset
<code>readOnlyRootFilesystem</code>	<code>boolean</code>	ReadOnlyRootFilesystem means the root filesyst read-only
<code>repository</code>	<code>string</code>	Repository defines the image repository from whi pull the MySQL server image.
<code>resources</code>	<code>object</code>	Resources holds ResourceRequirements for the Agent & Server Containers

Property	Type	Description
<code>rootPasswordSecret</code>	<code>object</code>	If defined, we use this secret for configuring the MYSQL_ROOT_PASSWORD. If it is not set we generate a secret dynamically.
<code>router</code>	<code>object</code>	Router defines the MySQL Router.
<code>runAsRoot</code>	<code>boolean</code>	RunAsRoot defines whether the MySQL container should run as root.
<code>securityContext</code>	<code>object</code>	SecurityContext holds Pod-level security attributes and common Container settings.
<code>slowSQL</code>	<code>object</code>	SlowSQL defines MySQL slow query log specification.
<code>sslSecret</code>	<code>object</code>	SSLSecret allows a user to specify custom CA certificate, server certificate and server key for Galera replication SSL.
<code>strictSecurityModeEnabled</code>	<code>boolean</code>	Enable this option to prevent sensitive information appearing in plain text in environment variables of a container.

Property	Type	Description
<code>tolerations</code>	<code>array</code>	Tolerations allows specifying a list of tolerations for controlling which set of Nodes a Pod can be scheduled on
<code>upgradeOption</code>	<code>object</code>	UpgradeOption defines the upgrade option for the MySQL cluster.
<code>useJemalloc</code>	<code>boolean</code>	UseJemalloc means use jemalloc for MySQL
<code>version</code>	<code>string</code>	Version defines the MySQL Docker image version
<code>volumeClaimTemplate</code>	<code>object</code>	VolumeClaimTemplate allows a user to specify how volumes inside a MySQL cluster

.spec.mgr.affinity

Description

If specified, affinity will define the pod's scheduling constraints

Type

`object`

Property	Type	Description
<code>nodeAffinity</code>	<code>object</code>	Describes node affinity scheduling rules for the pod.

Property	Type	Description
<code>podAffinity</code>	<code>object</code>	Describes pod affinity scheduling rules (e.g. co-locate this pod in the same node, zone, etc. as some other pod(s)).
<code>podAntiAffinity</code>	<code>object</code>	Describes pod anti-affinity scheduling rules (e.g. avoid putting this pod in the same node, zone, etc. as some other pod(s)).

`.spec.mgr.affinity.nodeAffinity`

Description

Describes node affinity scheduling rules for the pod.

Type

`object`

Property	Type	Description
<code>preferredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of th

Property	Type	Description
		scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc. compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest sum are the most preferred.
requiredDuringSchedulingIgnoredDuringExecution	object	If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to an update), the system may or may not try to eventually evict the pod from its node.

`.spec.mgr.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest sum are the most preferred.

Type

array

`.spec.mgr.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

An empty preferred scheduling term matches all objects with implicit weight 0 (i.e. it's a no-op). A null preferred scheduling term matches no objects (i.e. is also a no-op).

Type

object

Required

preference

weight

Property	Type	Description
<code>preference</code>	<code>object</code>	A node selector term, associated with the corresponding weight.

Property	Type	Description
<code>weight</code>	<code>integer</code>	Weight associated with matching the corresponding nodeSelectorTerm, in the range 1-100.

`.spec.mgr.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference`

Description

A node selector term, associated with the corresponding weight.

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	A list of node selector requirements by node's labels.
<code>matchFields</code>	<code>array</code>	A list of node selector requirements by node's fields.

`.spec.mgr.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions`

Description

A list of node selector requirements by node's labels.

Type

`array`

`.spec.mgr.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[]`

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

`.spec.mgr.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values`

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.mgr.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values[]

Type

string

.spec.mgr.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields

Description

A list of node selector requirements by node's fields.

Type

array

.spec.mgr.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[]

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist, Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

`.spec.mgr.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values`

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

`.spec.mgr.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values`

Type

`string`

`.spec.mgr.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution`

Description

If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to an update), the system may or may not try to eventually evict the pod from its node.

Type

`object`

Required

`nodeSelectorTerms`

Property	Type	Description
<code>nodeSelectorTerms</code>	<code>array</code>	Required. A list of node selector terms. The terms are ORed.

`.spec.mgr.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms`

Description

Required. A list of node selector terms. The terms are ORed.

Type

array

.spec.mgr.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[]

Description

A null or empty node selector term matches no objects. The requirements of them are ANDed. The TopologySelectorTerm type implements a subset of the NodeSelectorTerm.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	A list of node selector requirements by node's labels.
<code>matchFields</code>	array	A list of node selector requirements by node's fields.

.spec.mgr.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions

Description

A list of node selector requirements by node's labels.

Type

array

.spec.mgr.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpre

ssions[]

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.mgr.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type`array`

.spec.mgr.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values[]

Type`string`

.spec.mgr.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields

Description

A list of node selector requirements by node's fields.

Type`array`

.spec.mgr.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[]

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type`object`

Required

key

operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist, Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.mgr.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

`.spec.mgr.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values[]`

Type

`string`

`.spec.mgr.affinity.podAffinity`

Description

Describes pod affinity scheduling rules (e.g. co-locate this pod in the same node, zone, etc. as some other pod(s)).

Type

`object`

Property	Type	Description
<code>preferredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, <code>requiredDuringSchedulingIgnoredDuringExecution</code> affinity expressions, etc.),

Property	Type	Description
		compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	If the affinity requirement: specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirement: specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

`.spec.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

`array`

`.spec.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Type

`object`

Required

`podAffinityTerm``weight`

Property	Type	Description
<code>podAffinityTerm</code>	<code>object</code>	Required. A pod affinity term, associated with the corresponding weight.

Property	Type	Description
<code>weight</code>	<code>integer</code>	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

`.spec.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm`

Description

Required. A pod affinity term, associated with the corresponding weight.

Type

`object`

Required

`topologyKey`

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is

Property	Type	Description
		forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
mismatchLabelKeys	array	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
namespaceSelector	object	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
namespaces	array	namespaces specifies a static list of namespace names that the term applies to. The term is applied to

Property	Type	Description
		the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an

Property	Type	Description
		element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
<code>key</code>	<code>string</code>	key is the label key that the selector applies to.
<code>operator</code>	<code>string</code>	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
<code>values</code>	<code>array</code>	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]`

Type

`string`

`.spec.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

`object`

`.spec.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys`

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

`array`

`.spec.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKey`

s[]**Type**

string

.spec.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]

Type

string

.spec.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSele`

ctor.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This

array is replaced during a strategic merge patch.

Type

array

.spec.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]

Type

string

.spec.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

string

.spec.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Type

array

.spec.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running

Type

object

Required

topologyKey

Property	Type	Description
labelSelector	object	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
matchLabelKeys	array	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
mismatchLabelKeys	array	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys

Property	Type	Description
		and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
namespaceSelector	object	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
namespaces	array	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
topologyKey	string	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]
.values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]

Type

string

.spec.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

`array`

`.spec.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]`

Type

`string`

`.spec.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys`

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

`array`

`.spec.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]`

Type

`string`

`.spec.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpres`

sions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This

array is replaced during a strategic merge patch.

Type

array

.spec.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]

Type

string

.spec.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

`.spec.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]`

Type

`string`

`.spec.mgr.affinity.podAntiAffinity`

Description

Describes pod anti-affinity scheduling rules (e.g. avoid putting this pod in the same node, zone, etc. as some other pod(s)).

Type

`object`

Property	Type	Description
<code>preferredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, <code>requiredDuringSchedulingIgnoredDuringExecution</code> anti-affinity expressions, etc.), compute a sum by

Property	Type	Description
		iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

`.spec.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling anti-affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

`array`

`.spec.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Type

`object`

Required

`podAffinityTerm``weight`

Property	Type	Description
<code>podAffinityTerm</code>	<code>object</code>	Required. A pod affinity term, associated with the corresponding weight.

Property	Type	Description
<code>weight</code>	<code>integer</code>	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

`.spec.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm`

Description

Required. A pod affinity term, associated with the corresponding weight.

Type

`object`

Required

`topologyKey`

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is

Property	Type	Description
		forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
mismatchLabelKeys	array	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
namespaceSelector	object	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
namespaces	array	namespaces specifies a static list of namespace names that the term applies to. The term is applied to

Property	Type	Description
		the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	matchExpressions is a list of label selector requirements. The requirements are ANDed.

Property	Type	Description
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

`key`

`operator`

Property	Type	Description
<code>key</code>	<code>string</code>	key is the label key that the selector applies to.
<code>operator</code>	<code>string</code>	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
<code>values</code>	<code>array</code>	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

`array`

.spec.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]

Type

`string`

.spec.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

`object`

.spec.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

`array`

.spec.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabel

Keys[]

Type

string

.spec.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]

Type

string

.spec.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespace`

Selector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This

array is replaced during a strategic merge patch.

Type

array

.spec.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]

Type

string

.spec.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

string

.spec.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Type

array

.spec.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running

Type

object

Required

topologyKey

Property	Type	Description
labelSelector	object	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
matchLabelKeys	array	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
mismatchLabelKeys	array	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same

Property	Type	Description
		key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
namespaceSelector	object	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
namespaces	array	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
topologyKey	string	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
<code>key</code>	<code>string</code>	key is the label key that the selector applies to.
<code>operator</code>	<code>string</code>	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
<code>values</code>	<code>array</code>	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]

Type

string

.spec.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is

empty. The same key is forbidden to exist in both `matchLabelKeys` and `labelSelector`. Also, `matchLabelKeys` cannot be set when `labelSelector` isn't set. This is a beta field and requires enabling `MatchLabelKeysInPodAffinity` feature gate (enabled by default).

Type

array

`.spec.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]`

Type

string

`.spec.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys`

Description

`MismatchLabelKeys` is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both `mismatchLabelKeys` and `labelSelector`. Also, `mismatchLabelKeys` cannot be set when `labelSelector` isn't set. This is a beta field and requires enabling `MatchLabelKeysInPodAffinity` feature gate (enabled by default).

Type

array

`.spec.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]`

Type

string

`.spec.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
<code>key</code>	<code>string</code>	key is the label key that the selector applies to.
<code>operator</code>	<code>string</code>	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
<code>values</code>	<code>array</code>	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]

Type

string

.spec.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]**Type**

string

.spec.mgr.backupVolumeClaimTemplate**Description**

BackupVolumeClaimTemplate allows a user to specify a volume to temporarily store the data for a backup prior to it being shipped to object storage.

Type

object

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources

Property	Type	Description
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
metadata	ObjectMeta ↗	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata ↗
spec	object	spec defines the desired characteristics of a volume requested by a pod author. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims ↗
status	object	status represents the current information/status of a persistent volume claim. Read-only. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims ↗

.spec.mgr.backupVolumeClaimTemplate.spec

Description

spec defines the desired characteristics of a volume requested by a pod author. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims>

Type

object

Property	Type	Description
accessModes	array	accessModes contains the desired access modes volume should have. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1
dataSource	object	dataSource field can be used to specify either: - An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) - An existing PVC (PersistentVolumeClaim) If the provisioner or an external controller can support the specified data source, it will create a new volume based on the contents of the specified data source. When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If the namespace is specified, then dataSourceRef will not be copied to dataSource.
dataSourceRef	object	dataSourceRef specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a PersistentVolumeClaim object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replace the functionality of the dataSource field and as such

Property	Type	Description
		both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in dataSourceRef, dataSource isn't set to the same value and must be empty. There are three important differences between dataSource and dataSourceRef: • While dataSource only allows two specific types of objects, dataSourceRef allows any non-core object as well as PersistentVolumeClaim objects. • While dataSource ignores disallowed values (dropping them), dataSourceRef preserves all values, and generates an error if a disallowed value is specified. • While dataSource only allows local objects, dataSourceRef allows objects in any namespace (Beta) Using this field requires the AnyVolumeDataSource feature gate to be enabled (Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource feature gate to be enabled.
resources	object	resources represents the minimum resources the volume should have. If RecoverVolumeExpansionFailure feature is enabled, users are allowed to specify resource requirement that are lower than previous value but must still be higher than capacity recorded in the status field of claim. More info:

Property	Type	Description
		https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources ↗
<code>selector</code>	<code>object</code>	selector is a label query over volumes to consider binding.
<code>storageClassName</code>	<code>string</code>	storageClassName is the name of the StorageClass required by the claim. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#class-1 ↗

Property	Type	Description
<code>volumeAttributesClassName</code>	<code>string</code>	<code>volumeAttributesClassName</code> may be used to set the <code>VolumeAttributesClass</code> used by this claim. If specified, the CSI driver will create or update the volume with attributes defined in the corresponding <code>VolumeAttributesClass</code>. This has a different purpose than <code>storageClassName</code>, it can be changed after the claim is created. An empty string value means that <code>VolumeAttributesClass</code> will be applied to the claim if it's not allowed to reset this field to empty string once it is set. If unspecified and the <code>PersistentVolumeClaim</code> is unbound, the default <code>VolumeAttributesClass</code> will be used by the <code>persistentvolume</code> controller if it exists. If the resource referred to by <code>volumeAttributesClass</code> does not exist, this <code>PersistentVolumeClaim</code> will be set to <code>Pending</code> state, as reflected by the <code>modifyVolumeStatus</code> field, until such as a resource exists. More info: https://kubernetes.io/docs/concepts/storage/volume-attributes-classes/ [↗] (Beta) Using this field requires the <code>VolumeAttributesClass</code> feature gate to be enabled (off by default).
<code>volumeMode</code>	<code>string</code>	<code>volumeMode</code> defines what type of volume is required by the claim. Value of <code>Filesystem</code> is implied when not included in claim spec.
<code>volumeName</code>	<code>string</code>	<code>volumeName</code> is the binding reference to the <code>PersistentVolume</code> backing this claim.

`.spec.mgr.backupVolumeClaimTemplate.spec.accessModes`

Description

accessModes contains the desired access modes the volume should have. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1>

Type

array

`.spec.mgr.backupVolumeClaimTemplate.spec.accessModes[]`

Type

string

`.spec.mgr.backupVolumeClaimTemplate.spec.dataSource`

Description

dataSource field can be used to specify either: * An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) * An existing PVC (PersistentVolumeClaim) If the provisioner or an external controller can support the specified data source, it will create a new volume based on the contents of the specified data source. When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If the namespace is specified, then dataSourceRef will not be copied to dataSource.

Type

object

Required

kind

name

Property	Type	Description
<code>apiGroup</code>	<code>string</code>	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
<code>kind</code>	<code>string</code>	Kind is the type of resource being referenced
<code>name</code>	<code>string</code>	Name is the name of resource being referenced

.spec.mgr.backupVolumeClaimTemplate.spec.dataSourceRef

Description

`dataSourceRef` specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a `PersistentVolumeClaim` object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replace the functionality of the `dataSource` field and as such if both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in `dataSourceRef`, both fields (`dataSource` and `dataSourceRef`) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in `dataSourceRef`, `dataSource` isn't set to the same value and must be empty. There are three important differences between `dataSource` and `dataSourceRef`:

- * While `dataSource` only allows two specific types of objects, `dataSourceRef` allows any non-core object, as well as `PersistentVolumeClaim` objects.
- * While `dataSource` ignores disallowed values (dropping them), `dataSourceRef` preserves all values, and generates an error if a disallowed value is specified.
- * While `dataSource` only allows local objects, `dataSourceRef` allows objects in any namespaces.

(Beta) Using this field requires the `AnyVolumeDataSource` feature gate to be enabled.
 (Alpha) Using the namespace field of `dataSourceRef` requires the `CrossNamespaceVolumeDataSource` feature gate to be enabled.

Type

object

Required

kind

name

Property	Type	Description
apiGroup	string	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
kind	string	Kind is the type of resource being referenced
name	string	Name is the name of resource being referenced
namespace	string	Namespace is the namespace of resource being referenced Note that when a namespace is specified, a gateway.networking.k8s.io/ReferenceGrant object is required in the referent namespace to allow that namespace's owner to accept the reference. See the ReferenceGrant documentation for details. (Alpha) This field requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

.spec.mgr.backupVolumeClaimTemplate.spec.resources

Description

resources represents the minimum resources the volume should have. If RecoverVolumeExpansionFailure feature is enabled users are allowed to specify resource

requirements that are lower than previous value but must still be higher than capacity recorded in the status field of the claim. More info:

<https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources>

Type

object

Property	Type	Description
<code>limits</code>	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗
<code>requests</code>	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

`.spec.mgr.backupVolumeClaimTemplate.spec.resources.limits`

Description

Limits describes the maximum amount of compute resources allowed. More info:
<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

.spec.mgr.backupVolumeClaimTemplate.spec.resources.requests

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info:

<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

.spec.mgr.backupVolumeClaimTemplate.spec.selector

Description

selector is a label query over volumes to consider for binding.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.mgr.backupVolumeClaimTemplate.spec.selector.matchExpressions

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.mgr.backupVolumeClaimTemplate.spec.selector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.

Property	Type	Description
<code>values</code>	<code>array</code>	<code>values</code> is an array of string values. If the operator is <code>In</code> or <code>NotIn</code> , the values array must be non-empty. If the operator is <code>Exists</code> or <code>DoesNotExist</code> , the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.mgr.backupVolumeClaimTemplate.spec.selector.matchExpressions[].values`

Description

`values` is an array of string values. If the operator is `In` or `NotIn`, the values array must be non-empty. If the operator is `Exists` or `DoesNotExist`, the values array must be empty. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.mgr.backupVolumeClaimTemplate.spec.selector.matchExpressions[].values[]`

Type

`string`

`.spec.mgr.backupVolumeClaimTemplate.spec.selector.matchLabels`

Description

`matchLabels` is a map of {key,value} pairs. A single {key,value} in the `matchLabels` map is equivalent to an element of `matchExpressions`, whose key field is "key", the operator is `In`, and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mgr.backupVolumeClaimTemplate.status

Description

status represents the current information/status of a persistent volume claim. Read-only.
More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims>

Type

object

Property	Type	Description
<code>accessModes</code>	array	accessModes contains the actual access modes of the volume backing the PVC has. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1
<code>allocatedResourceStatuses</code>	object	allocatedResourceStatuses stores status of resources being resized for the given PVC. Key names must follow standard Kubernetes label syntax. Valid values are either: * Un-prefixed keys: - storage - the resource of the volume. * Custom resources must use implementation-defined prefixed names such as "example.com/my-custom-resource" Apart from reserved values - keys that are unprefixed or have kubernetes.io prefix are considered reserved and hence may not be used. ClaimResourceStatus can be in any of following states: - ControllerResizeInProgress: State set when resize controller starts resizing the volume plane. - ControllerResizeFailed: State set when resize has failed in resize controller with a termin

Property	Type	Description
		NodeResizePending: State set when resize has finished resizing the volume but further volume is needed on the node. - NodeResizeInProgress: State set when kubelet is resizing the volume. - NodeResizeFailed: when resizing has failed in kubelet with a permanent error. Transient errors don't set NodeResizeFailed. example: if expanding a PVC for more capacity, the pvc.status.allocatedResourceStatus['storageClassName'] field can be one of the following states: - pvc.status.allocatedResourceStatus['storageClassName'] = "ControllerResizeInProgress" - pvc.status.allocatedResourceStatus['storageClassName'] = "ControllerResizeFailed" - pvc.status.allocatedResourceStatus['storageClassName'] = "NodeResizePending" - pvc.status.allocatedResourceStatus['storageClassName'] = "NodeResizeInProgress" - pvc.status.allocatedResourceStatus['storageClassName'] = "NodeResizeFailed" When this field is not set, it means that no resize operation is in progress for the given PVC. A controller that receives PVC update with unknown resourceName or ClaimResourceName should ignore the update for the purpose it was designed. For example - a controller that is responsible for resizing capacity of the volume should ignore PVC updates that change other values associated with PVC. This is an alpha field and requires enabling the RecoverVolumeExpansionFailure feature.

Property	Type	Description
<code>allocatedResources</code>	object	<code>allocatedResources</code> tracks the resources allocated to a PVC including its capacity. Key names follow standard Kubernetes label syntax. Valid values are either: * Un-prefixed keys: - storage - the capacity of the volume. * Custom resources must use implementation-defined prefixed names such as "example.com/my-custom-resource". Annotations with keys that are unprefixed or have the <code>kubernetes.io</code> prefix are considered reserved and hence may not be used. Capacity reported here may be larger than the actual capacity when a volume expansion operation is requested. For storage quota, the larger value of <code>allocatedResources</code> and <code>PVC.spec.resources.requests.storage</code> is used for quota calculation. If <code>allocatedResources</code> is not set, <code>PVC.spec.resources.requests.storage</code> alone is used for quota calculation. If a volume expansion capacity request is lowered, <code>allocatedResources</code> is only lowered if there are no expansion operations in progress and if the volume capacity is equal or lower than the requested capacity. A controller that receives PVC update with an unknown <code>resourceName</code> should ignore the update for the purpose it was designed. For example, a controller that only is responsible for resizing the volume, should ignore PVC updates that change other valid resources associated with the volume. This is an alpha field and requires enabling the <code>RecoverVolumeExpansionFailure</code> feature.

Property	Type	Description
capacity	object	capacity represents the actual resources consumed by the underlying volume.
conditions	array	conditions is the current Condition of persistentVolumeClaim. If underlying persistent volume is resized then the Condition will be set to 'Resizing'.
currentVolumeAttributesClassName	string	currentVolumeAttributesClassName is the name of the VolumeAttributesClass the PersistentVolumeClaim is requesting to be used to create the PersistentVolume. When unset, there is no VolumeAttributesClass requested for this PersistentVolumeClaim. This is a beta feature and requires enabling VolumeAttributesClass feature (off by default).
modifyVolumeStatus	object	ModifyVolumeStatus represents the status of the ControllerModifyVolume operation. When there is no ModifyVolume operation being performed by the controller, this field is not set. This is a beta field and requires enabling VolumeAttributesClass feature (off by default).
phase	string	phase represents the current phase of PersistentVolumeClaim.

.spec.mgr.backupVolumeClaimTemplate.status.accessModes

Description

accessModes contains the actual access modes the volume backing the PVC has. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1>

Type

array

.spec.mgr.backupVolumeClaimTemplate.status.accessModes[]

Type

string

.spec.mgr.backupVolumeClaimTemplate.status.allocatedResourceStatuses

Description

allocatedResourceStatuses stores status of resource being resized for the given PVC. Key names follow standard Kubernetes label syntax. Valid values are either: * Un-prefixed keys: - storage - the capacity of the volume. * Custom resources must use implementation-defined prefixed names such as "example.com/my-custom-resource" Apart from above values - keys that are unprefixed or have kubernetes.io prefix are considered reserved and hence may not be used. ClaimResourceStatus can be in any of following states: -

ControllerResizeInProgress: State set when resize controller starts resizing the volume in control-plane. - ControllerResizeFailed: State set when resize has failed in resize controller with a terminal error. - NodeResizePending: State set when resize controller has finished resizing the volume but further resizing of volume is needed on the node. -

NodeResizeInProgress: State set when kubelet starts resizing the volume. -

NodeResizeFailed: State set when resizing has failed in kubelet with a terminal error.

Transient errors don't set NodeResizeFailed. For example: if expanding a PVC for more capacity - this field can be one of the following states: -

pvc.status.allocatedResourceStatus['storage'] = "ControllerResizeInProgress" -

pvc.status.allocatedResourceStatus['storage'] = "ControllerResizeFailed" -

pvc.status.allocatedResourceStatus['storage'] = "NodeResizePending" -

pvc.status.allocatedResourceStatus['storage'] = "NodeResizeInProgress" -

pvc.status.allocatedResourceStatus['storage'] = "NodeResizeFailed" When this field is not set, it means that no resize operation is in progress for the given PVC. A controller that

receives PVC update with previously unknown resourceName or ClaimResourceStatus should ignore the update for the purpose it was designed. For example - a controller that only is responsible for resizing capacity of the volume, should ignore PVC updates that change other valid resources associated with PVC. This is an alpha field and requires enabling RecoverVolumeExpansionFailure feature.

Type

object

.spec.mgr.backupVolumeClaimTemplate.status.allocatedResources

Description

allocatedResources tracks the resources allocated to a PVC including its capacity. Key names follow standard Kubernetes label syntax. Valid values are either: * Un-prefixed keys: - storage - the capacity of the volume. * Custom resources must use implementation-defined prefixed names such as "example.com/my-custom-resource" Apart from above values - keys that are unprefixed or have kubernetes.io prefix are considered reserved and hence may not be used. Capacity reported here may be larger than the actual capacity when a volume expansion operation is requested. For storage quota, the larger value from allocatedResources and PVC.spec.resources is used. If allocatedResources is not set, PVC.spec.resources alone is used for quota calculation. If a volume expansion capacity request is lowered, allocatedResources is only lowered if there are no expansion operations in progress and if the actual volume capacity is equal or lower than the requested capacity. A controller that receives PVC update with previously unknown resourceName should ignore the update for the purpose it was designed. For example - a controller that only is responsible for resizing capacity of the volume, should ignore PVC updates that change other valid resources associated with PVC. This is an alpha field and requires enabling RecoverVolumeExpansionFailure feature.

Type

object

.spec.mgr.backupVolumeClaimTemplate.status.capacity

Description

capacity represents the actual resources of the underlying volume.

Type

object

.spec.mgr.backupVolumeClaimTemplate.status.conditions

Description

conditions is the current Condition of persistent volume claim. If underlying persistent volume is being resized then the Condition will be set to 'Resizing'.

Type

array

.spec.mgr.backupVolumeClaimTemplate.status.conditions

[]

Description

PersistentVolumeClaimCondition contains details about state of pvc

Type

object

Required

status

type

Property	Type	Description
lastProbeTime	string	lastProbeTime is the time we probed the condition.
lastTransitionTime	string	lastTransitionTime is the time the condition transitioned fro

Property	Type	Description
<code>message</code>	<code>string</code>	message is the human-readable message indicating detail
<code>reason</code>	<code>string</code>	reason is a unique, this should be a short, machine unders the reason for condition's last transition. If it reports "Resizi underlying persistent volume is being resized.
<code>status</code>	<code>string</code>	Status is the status of the condition. Can be True, False, U https://kubernetes.io/docs/reference/kubernetes-api/config-resources/persistent-volume-claim-v1/#:~:text=state%20of%20pvc-,conditions.status,-(string)%
<code>type</code>	<code>string</code>	Type is the type of the condition. More info: https://kubernetes.io/docs/reference/kubernetes-api/config-resources/persistent-volume-claim-v1/#:~:text=set%20to%20%27ResizeStarted%27.-,Persiste contains%20details%20about

`.spec.mgr.backupVolumeClaimTemplate.status.modifyVolumeStatus`

Description

ModifyVolumeStatus represents the status object of ControllerModifyVolume operation. When this is unset, there is no ModifyVolume operation being attempted. This is a beta field and requires enabling VolumeAttributesClass feature (off by default).

Type

`object`

Required

status

Property	Type	Description
status	string	status is the status of the ControllerModifyVolume operation. It can be in any of following states: • Pending Pending indicates that the PersistentVolumeClaim cannot be modified due to unmet requirements, such as the specified VolumeAttributesClass not existing. • InProgress InProgress indicates that the volume is being modified. • Infeasible Infeasible indicates that the request has been rejected as invalid by the CSI driver. To resolve the error, a valid VolumeAttributesClass needs to be specified. Note: New statuses can be added in the future. Consumers should check for unknown statuses and fail appropriately.
targetVolumeAttributesClassName	string	targetVolumeAttributesClassName is the name of the VolumeAttributesClass the PVC currently being reconciled

Description

CertConfig allows a user to specify custom CA certificate with advanced option.

Type

object

Property	Type	Description
DNSNames	array	DNSNames means the dns name list of the cert.
duration	string	Duration means the duration of the cert.
secretName	string	SecretName means the secret of the cert.

.spec.mgr.certConfig.DNSNames

Description

DNSNames means the dns name list of the cert.

Type

array

.spec.mgr.certConfig.DNSNames[]

Type

string

.spec.mgr.config

Description

Config allows a user to specify a custom configuration file for MySQL.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

.spec.mgr.image

Description

Image allows a user to specify image.

Type

object

Property	Type	Description
agentImage	string	AgentImage means the image of the agent.
routerImage	string	RouterImage means the image of the router.
serverImage	string	ServerImage means the image of the server.
slowSQL	string	SlowSQL means the image of the slow sql.

.spec.mgr.imagePullSecret

Description

ImagePullSecret defines the name of the secret that contains the required credentials for pulling from the specified Repository.

Type

array

.spec.mgr.imagePullSecret[]

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

.spec.mgr.monitor

Description

Monitor define MySQL monitor spec

Type

object

Property	Type	Description
<code>enable</code>	<code>boolean</code>	Enable means enable monitor
<code>exporter</code>	<code>object</code>	Exporter define MySQL exporter spec

`.spec.mgr.monitor.exporter`

Description

Exporter define MySQL exporter spec

Type

`object`

Property	Type	Description
<code>image</code>	<code>string</code>	Image means the image of the exporter.
<code>resources</code>	<code>object</code>	Resources holds ResourceRequirements for the MySQL Agent & Server Containers.

`.spec.mgr.monitor.exporter.resources`

Description

Resources holds ResourceRequirements for the MySQL Agent & Server Containers.

Type

`object`

Property	Type	Description
		Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container.
claims	array	This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
limits	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗
requests	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

.spec.mgr.monitor.exporter.resources.claims

Description

Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.

Type

array

.spec.mgr.monitor.exporter.resources.claims[]

Description

ResourceClaim references one entry in PodSpec.ResourceClaims.

Type

object

Required

name

Property	Type	Description
name	string	Name must match the name of one entry in pod.spec.resourceClaims of the Pod where this field is used. It makes that resource available inside a container.
request	string	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

.spec.mgr.monitor.exporter.resources.limits

Description

Limits describes the maximum amount of compute resources allowed. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

.spec.mgr.monitor.exporter.resources.requests

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info:

<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

.spec.mgr.multiClusterReplication

Description

Config this field to support multi cluster replication for disaster recovery.

Type

object

Property	Type	Description
<code>bootstrapSecret</code>	<code>string</code>	BootstrapSecret means the secret of the multi cluster replication.
<code>clusterSetSeedAddr</code>	<code>string</code>	ClusterSetSeedAddr means the cluster set seed address of the multi cluster replication.
<code>enabled</code>	<code>boolean</code>	Enabled means enable multi cluster replication
<code>hostPort</code>	<code>integer</code>	HostPort means the host port of the multi cluster replication.

.spec.mgr.nodeSelector

Description

NodeSelector is a selector which must be true for the pod to fit on a node. Selector which must match a node's labels for the pod to be scheduled on that node. More info: <https://kubernetes.io/docs/concepts/configuration/assign-pod-node/>

Type

object

.spec.mgr.paras

Description

Paras provide paras template feature

Type

object

.spec.mgr.patches

Description

Patches use to apply patches to the MySQL container

Type

object

Property	Type	Description
podPatches	object	PodPatches is a map of pod name to patch
servicePatches	object	ServicePatches is a map of service name to patch

.spec.mgr.patches.podPatches

Description

PodPatches is a map of pod name to patch

Type

object

.spec.mgr.patches.servicePatches

Description

ServicePatches is a map of service name to patch

Type

object

.spec.mgr.resources

Description

Resources holds ResourceRequirements for the MySQL Agent & Server Containers

Type

object

Property	Type	Description
agent	object	Agent describes the compute resource requirements for agent.
server	object	Server describes the compute resource requirements for server.

.spec.mgr.resources.agent

Description

Agent describes the compute resource requirements for agent.

Type

object

Property	Type	Description
		Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container.
claims	array	This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
limits	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗
requests	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

.spec.mgr.resources.agent.claims

Description

Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.

Type

array

.spec.mgr.resources.agent.claims[]

Description

ResourceClaim references one entry in PodSpec.ResourceClaims.

Type

object

Required

name

Property	Type	Description
name	string	Name must match the name of one entry in pod.spec.resourceClaims of the Pod where this field is used. It makes that resource available inside a container.
request	string	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

.spec.mgr.resources.agent.limits

Description

Limits describes the maximum amount of compute resources allowed. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

.spec.mgr.resources.agent.requests

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info:

<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

.spec.mgr.resources.server

Description

Server describes the compute resource requirements for server.

Type

object

Property	Type	Description
claims	array	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
limits	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗
requests	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to

Property	Type	Description
		Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

.spec.mgr.resources.server.claims

Description

Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.

Type

array

.spec.mgr.resources.server.claims[]

Description

ResourceClaim references one entry in PodSpec.ResourceClaims.

Type

object

Required

name

Property	Type	Description
name	string	Name must match the name of one entry in pod.spec.resourceClaims of the Pod where this field is used. It makes that resource available inside a container.

Property	Type	Description
<code>request</code>	<code>string</code>	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

.spec.mgr.resources.server.limits

Description

Limits describes the maximum amount of compute resources allowed. More info:
<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

.spec.mgr.resources.server.requests

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info:
<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

.spec.mgr.rootPasswordSecret

Description

If defined, we use this secret for configuring the MYSQL_ROOT_PASSWORD If it is not set we generate a secret dynamically

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

.spec.mgr.router

Description

Router defines the MySQL Router

Type

`object`

Property	Type	Description
<code>affinity</code>	<code>object</code>	If specified, affinity will define the pod's scheduling constraints
<code>configuration</code>	<code>string</code>	Configuration means the configuration of the router.
<code>mysqlPasswordSecret</code>	<code>string</code>	MySQLPasswordSecret defines the secret name of MySQL user.
<code>mysqlUser</code>	<code>string</code>	MySQLUser defines the user that connect to MySQL.

Property	Type	Description
<code>nodeSelector</code>	<code>object</code>	NodeSelector is a selector which must be true for the pod to fit on a node. Selector which must match a node's labels for the pod to be scheduled on that node. More info: https://kubernetes.io/docs/concepts/configuration/assign-pod-node/ ↗
<code>replicas</code>	<code>integer</code>	Replicas defines the replicas of the MySQL Router.
<code>resources</code>	<code>object</code>	Resources holds ResourceRequirements for the MySQL Agent & Server Containers
<code>securityContext</code>	<code>object</code>	PodSecurityContext holds pod-level security attributes and common container settings. Some fields are also present in container.securityContext. Field values of container.securityContext take precedence over field values of PodSecurityContext.
<code>svcRO</code>	<code>object</code>	SvcRo defines the read only service of the MySQL Router.
<code>svcRW</code>	<code>object</code>	SvcRW defines the read write service of the MySQL Router.
<code>tolerations</code>	<code>array</code>	Tolerations allows specifying a list of tolerations for controlling which set of Nodes a Pod can be scheduled

Property	Type	Description
		on If specified, configuration will define the router's configuration.

.spec.mgr.router.affinity

Description

If specified, affinity will define the pod's scheduling constraints

Type

object

Property	Type	Description
<code>nodeAffinity</code>	object	Describes node affinity scheduling rules for the pod.
<code>podAffinity</code>	object	Describes pod affinity scheduling rules (e.g. co-locate this pod in the same node, zone, etc. as some other pod(s)).
<code>podAntiAffinity</code>	object	Describes pod anti-affinity scheduling rules (e.g. avoid putting this pod in the same node, zone, etc. as some other pod(s)).

.spec.mgr.router.affinity.nodeAffinity

Description

Describes node affinity scheduling rules for the pod.

Type

object

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution affinity expressions, etc.) compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest sum are the most preferred.
requiredDuringSchedulingIgnoredDuringExecution	object	If the affinity requirements specified by this field are not met at scheduling

Property	Type	Description
		time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to an update), the system may or may not try to eventually evict the pod from its node.

`.spec.mgr.router.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest sum are the most preferred.

Type

array

`.spec.mgr.router.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

An empty preferred scheduling term matches all objects with implicit weight 0 (i.e. it's a no-op). A null preferred scheduling term matches no objects (i.e. is also a no-op).

Type

object

Required

preference

weight

Property	Type	Description
preference	object	A node selector term, associated with the corresponding weight.
weight	integer	Weight associated with matching the corresponding nodeSelectorTerm, in the range 1-100.

.spec.mgr.router.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference

Description

A node selector term, associated with the corresponding weight.

Type

object

Property	Type	Description
matchExpressions	array	A list of node selector requirements by node's labels.
matchFields	array	A list of node selector requirements by node's fields.

`.spec.mgr.router.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions`

Description

A list of node selector requirements by node's labels.

Type

array

`.spec.mgr.router.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[]`

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
<code>key</code>	<code>string</code>	The label key that the selector applies to.
<code>operator</code>	<code>string</code>	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.

Property	Type	Description
<code>values</code>	<code>array</code>	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

`.spec.mgr.router.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values`

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.mgr.router.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values[]`

Type

`string`

`.spec.mgr.router.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields`

Description

A list of node selector requirements by node's fields.

Type

array

`.spec.mgr.router.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[]`

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
<code>key</code>	<code>string</code>	The label key that the selector applies to.
<code>operator</code>	<code>string</code>	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
<code>values</code>	<code>array</code>	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.mgr.router.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.mgr.router.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values[]

Type

string

.spec.mgr.router.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to an update), the system may or may not try to eventually evict the pod from its node.

Type

object

Required

nodeSelectorTerms

Property	Type	Description
<code>nodeSelectorTerms</code>	<code>array</code>	Required. A list of node selector terms. The terms are ORed.

`.spec.mgr.router.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms`

Description

Required. A list of node selector terms. The terms are ORed.

Type

`array`

`.spec.mgr.router.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[]`

Description

A null or empty node selector term matches no objects. The requirements of them are ANDed. The TopologySelectorTerm type implements a subset of the NodeSelectorTerm.

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	A list of node selector requirements by node's labels.
<code>matchFields</code>	<code>array</code>	A list of node selector requirements by node's fields.

`.spec.mgr.router.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions`

Description

A list of node selector requirements by node's labels.

Type

array

`.spec.mgr.router.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[]`

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
<code>key</code>	<code>string</code>	The label key that the selector applies to.
<code>operator</code>	<code>string</code>	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.

Property	Type	Description
<code>values</code>	<code>array</code>	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

`.spec.mgr.router.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values`

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.mgr.router.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values[]`

Type

`string`

`.spec.mgr.router.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values[]`

hFields

Description

A list of node selector requirements by node's fields.

Type

array

.spec.mgr.router.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[]

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will

Property	Type	Description
		be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.mgr.router.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.mgr.router.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values[]

Type

string

.spec.mgr.router.affinity.podAffinity

Description

Describes pod affinity scheduling rules (e.g. co-locate this pod in the same node, zone, etc. as some other pod(s)).

Type

object

Property	Type	Description
<code>preferredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, <code>requiredDuringSchedulingIgnoredDuringExecution</code>, affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding <code>podAffinityTerm</code>; the node(s) with the highest sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	If the affinity requirements specified by this field are not met at scheduling time, the pod will not be

Property	Type	Description
		scheduled onto the node. If the affinity requirement: specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

.spec.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

array

`.spec.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

The weights of all of the matched `WeightedPodAffinityTerm` fields are added per-node to find the most preferred node(s)

Type

object

Required

podAffinityTerm

weight

Property	Type	Description
<code>podAffinityTerm</code>	object	Required. A pod affinity term, associated with the corresponding weight.
<code>weight</code>	integer	weight associated with matching the corresponding <code>podAffinityTerm</code> , in the range 1-100.

`.spec.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm`

Description

Required. A pod affinity term, associated with the corresponding weight.

Type

object

Required

topologyKey

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field

Property	Type	Description
		and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

.spec.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSele

ctor

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

object

Property	Type	Description
matchExpressions	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
matchLabels	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSele

ctor.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This

array is replaced during a strategic merge patch.

Type

array

.spec.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]

Type

string

.spec.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti)

affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both `matchLabelKeys` and `labelSelector`. Also, `matchLabelKeys` cannot be set when `labelSelector` isn't set. This is a beta field and requires enabling `MatchLabelKeysInPodAffinity` feature gate (enabled by default).

Type

array

`.spec.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]`

Type

string

`.spec.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys`

Description

`MismatchLabelKeys` is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both `mismatchLabelKeys` and `labelSelector`. Also, `mismatchLabelKeys` cannot be set when `labelSelector` isn't set. This is a beta field and requires enabling `MatchLabelKeysInPodAffinity` feature gate (enabled by default).

Type

array

`.spec.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatch`

LabelKeys[]

Type

string

.spec.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespa

ceSelector.matchExpressions

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.

Property	Type	Description
<code>values</code>	<code>array</code>	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]`

Type

`string`

`.spec.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

string

.spec.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may

or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Type

array

.spec.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running

Type

object

Required

topologyKey

Property	Type	Description
labelSelector	object	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
matchLabelKeys	array	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity.

Property	Type	Description
		Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Property	Type	Description
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	matchExpressions is a list of label selector requirements. The requirements are ANDed.

Property	Type	Description
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

`key`

`operator`

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]

Type

`string`

`.spec.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

`object`

`.spec.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys`

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

`array`

`.spec.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]`

Type

`string`

`.spec.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys`

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

`.spec.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]`

Type

string

`.spec.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.match

Expressions[].values[]

Type

string

.spec.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]

Type

string

.spec.mgr.router.affinity.podAntiAffinity

Description

Describes pod anti-affinity scheduling rules (e.g. avoid putting this pod in the same node, zone, etc. as some other pod(s)).

Type

object

Property	Type	Description
<code>preferredDuringSchedulingIgnoredDuringExecution</code>	array	The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, <code>requiredDuringSchedulingIgnoredDuringExecution</code> , anti-affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding <code>podAffinityTerm</code> ; the

Property	Type	Description
		node(s) with the highest sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

`.spec.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the

expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling anti-affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

array

.spec.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

Description

The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Type

object

Required

podAffinityTerm

weight

Property	Type	Description
<code>podAffinityTerm</code>	<code>object</code>	Required. A pod affinity term, associated with the corresponding weight.
<code>weight</code>	<code>integer</code>	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

.spec.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm

Description

Required. A pod affinity term, associated with the corresponding weight.

Type

object

Required

topologyKey

Property	Type	Description
labelSelector	object	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
matchLabelKeys	array	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
mismatchLabelKeys	array	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with

Property	Type	Description
		<code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both <code>mismatchLabelKeys</code> and <code>labelSelector</code> . Also, <code>mismatchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. This is a beta field and requires enabling <code>MatchLabelKeysInPodAffinity</code> feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the <code>namespaces</code> field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector (<code>{}</code>) matches all namespaces.
<code>namespaces</code>	<code>array</code>	<code>namespaces</code> specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by <code>namespaceSelector</code> . null or empty namespaces list and null <code>namespaceSelector</code> means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the <code>labelSelector</code> in the specified namespaces, where co-located is defined as running on a node whose value of the label with key <code>topologyKey</code> matches that of any

Property	Type	Description
		node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]

Type

string

.spec.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

`.spec.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys`

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

`.spec.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]`

Type

string

`.spec.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys`

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti)

affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both `mismatchLabelKeys` and `labelSelector`. Also, `mismatchLabelKeys` cannot be set when `labelSelector` isn't set. This is a beta field and requires enabling `MatchLabelKeysInPodAffinity` feature gate (enabled by default).

Type

array

`.spec.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]`

Type

string

`.spec.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the `namespaces` field. A null selector and null or empty namespaces list means "this pod's namespace". An empty selector (`{}`) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.

Property	Type	Description
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces.selector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces.selector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

`key`

`operator`

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces.selector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces.selector.matchExpressions[].values[]

Type

string

.spec.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

string

`.spec.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution`

Description

If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each `podAffinityTerm` are intersected, i.e. all terms must be satisfied.

Type

`array`

`.spec.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[]`

Description

Defines a set of pods (namely those matching the `labelSelector` relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key `<topologyKey>` matches that of any node on which a pod of the set of pods is running

Type

`object`

Required

`topologyKey`

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	A label query over a set of resources, in this case pods. If it's null, this <code>PodAffinityTerm</code> matches with no Pods.

Property	Type	Description
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Property	Type	Description
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector (<code>{}</code>) matches all namespaces.
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]`

Type

string

`.spec.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

`.spec.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys`

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

`.spec.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]`

Type

string

`.spec.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys`

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

`.spec.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]`

Type

string

`.spec.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]

Type

string

.spec.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

`.spec.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]`

Type

`string`

`.spec.mgr.router.nodeSelector`

Description

NodeSelector is a selector which must be true for the pod to fit on a node. Selector which must match a node's labels for the pod to be scheduled on that node. More info:

<https://kubernetes.io/docs/concepts/configuration/assign-pod-node/>

Type

`object`

`.spec.mgr.router.resources`

Description

Resources holds ResourceRequirements for the MySQL Agent & Server Containers

Type

`object`

Property	Type	Description
		Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container.
<code>claims</code>	<code>array</code>	This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.

Property	Type	Description
<code>limits</code>	<code>object</code>	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗
<code>requests</code>	<code>object</code>	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

`.spec.mgr.router.resources.claims`

Description

Claims lists the names of resources, defined in `spec.resourceClaims`, that are used by this container. This is an alpha field and requires enabling the `DynamicResourceAllocation` feature gate. This field is immutable. It can only be set for containers.

Type

`array`

`.spec.mgr.router.resources.claims[]`

Description

ResourceClaim references one entry in `PodSpec.ResourceClaims`.

Type

`object`

Required

name

Property	Type	Description
name	string	Name must match the name of one entry in pod.spec.resourceClaims of the Pod where this field is used. It makes that resource available inside a container.
request	string	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

.spec.mgr.router.resources.limits

Description

Limits describes the maximum amount of compute resources allowed. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

.spec.mgr.router.resources.requests

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

.spec.mgr.router.securityContext

Description

PodSecurityContext holds pod-level security attributes and common container settings. Some fields are also present in container.securityContext. Field values of container.securityContext take precedence over field values of PodSecurityContext.

Type

object

Property	Type	Description
appArmorProfile	object	appArmorProfile is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.
fsGroup	integer	A special supplemental group that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod: 1. The owning GID will be the FSGroup 2. The setgid bit is set (new files created in the volume will be owned by FSGroup) 3. The permission bits are OR'd with rw-rw-- -- If unset, the Kubelet will not modify the ownership and permissions of any volume. Note that this field cannot be set when spec.os.name is windows.
fsGroupChangePolicy	string	fsGroupChangePolicy defines behavior of changing ownership and permission of the volume before being exposed inside Pod.

Property	Type	Description
		This field will only apply to volume types which support fsGroup based ownership (and permissions). It will have no effect on ephemeral volume types such as: secret, configmaps and emptydir. Valid values are "OnRootMismatch" and "Always". If not specified, "Always" is used. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

Property	Type	Description
<code>runAsUser</code>	<code>integer</code>	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
<code>seLinuxChangePolicy</code>	<code>string</code>	<code>seLinuxChangePolicy</code> defines how the container's SELinux label is applied to all volumes used by the Pod. It has no effect on nodes that do not support SELinux or to volumes does not support SELinux. Valid values are "MountOption" and "Recursive". "Recursive" means relabeling of all files on all Pod volumes by the container runtime. This may be slow for large volumes, but allows mixing privileged and unprivileged Pods sharing the same volume on the same node. "MountOption" mounts all eligible Pod volumes with <code>-o context</code> mount option. This requires all Pods that share the same volume to use the same SELinux label. It is not possible to share the same volume among privileged and unprivileged Pods. Eligible volumes are in-tree FibreChannel and iSCSI volumes, and all CSI volumes whose CSI driver announces SELinux

Property	Type	Description
		support by setting <code>spec.seLinuxMount: true</code> in their <code>CSIDriver</code> instance. Other volumes are always re-labelled recursively. "MountOption" value is allowed only when SELinuxMount feature gate is enabled. If not specified and SELinuxMount feature gate is enabled, "MountOption" is used. If not specified and SELinuxMount feature gate is disabled, "MountOption" is used for <code>ReadWriteOncePod</code> volumes and "Recursive" for all other volumes. This field affects only Pods that have SELinux label set, either in <code>PodSecurityContext</code> or in <code>SecurityContext</code> of all containers. All Pods that use the same volume should use the same <code>seLinuxChangePolicy</code>, otherwise some pods can get stuck in <code>ContainerCreating</code> state. Note that this field cannot be set when <code>spec.os.name</code> is <code>windows</code>.

Property	Type	Description
<code>seLinuxOptions</code>	<code>object</code>	The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
<code>seccompProfile</code>	<code>object</code>	The seccomp options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
<code>supplementalGroups</code>	<code>array</code>	A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.
<code>supplementalGroupsPolicy</code>	<code>string</code>	Defines how supplemental groups of the first container processes are calculated. Valid values are "Merge" and "Strict". If not specified, "Merge" is used. (Alpha) Using the field requires the SupplementalGroupsPolicy feature gate to be enabled and the container runtime must implement support for this feature. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
<code>sysctls</code>	<code>array</code>	Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.
<code>windowsOptions</code>	<code>object</code>	The Windows specific settings applied to all containers. If unspecified, the options within a container's SecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

`.spec.mgr.router.securityContext.appArmorProfile`

Description

appArmorProfile is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured

Property	Type	Description
		on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
<code>type</code>	<code>string</code>	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

.spec.mgr.router.securityContext.seLinuxOptions

Description

The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Property	Type	Description
<code>level</code>	<code>string</code>	Level is SELinux level label that applies to the container.
<code>role</code>	<code>string</code>	Role is a SELinux role label that applies to the container.
<code>type</code>	<code>string</code>	Type is a SELinux type label that applies to the container.

Property	Type	Description
<code>user</code>	<code>string</code>	User is a SELinux user label that applies to the container.

`.spec.mgr.router.securityContext.seccompProfile`

Description

The seccomp options to use by the containers in this pod. Note that this field cannot be set when `spec.os.name` is windows.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.
<code>type</code>	<code>string</code>	<code>type</code> indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied.

.spec.mgr.router.securityContext.supplementalGroups

Description

A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.

Type

array

.spec.mgr.router.securityContext.supplementalGroups[]

Type

integer

.spec.mgr.router.securityContext.sysctls

Description

Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.

Type

array

.spec.mgr.router.securityContext.sysctls[]

Description

Sysctl defines a kernel parameter to be set

Type

object

Required

name

value

Property	Type	Description
name	string	Name of a property to set
value	string	Value of a property to set

.spec.mgr.router.securityContext.windowsOptions

Description

The Windows specific settings applied to all containers. If unspecified, the options within a container's SecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
gmsaCredentialSpec	string	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.

Property	Type	Description
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entryptoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

.spec.mgr.router.svcRO

Description

SvcRo defines the read only service of the MySQL Router.

Type

`object`

Property	Type	Description
<code>port</code>	<code>integer</code>	Port means the port of the node port.
<code>type</code>	<code>string</code>	Type means the type of the service.

`.spec.mgr.router.svcRW`

Description

SvcRW defines the read write service of the MySQL Router.

Type

`object`

Property	Type	Description
<code>port</code>	<code>integer</code>	Port means the port of the node port.
<code>type</code>	<code>string</code>	Type means the type of the service.

`.spec.mgr.router.tolerations`

Description

Tolerations allows specifying a list of tolerations for controlling which set of Nodes a Pod can be scheduled on. If specified, configuration will define the router's configuration.

Type

`array`

.spec.mgr.router.tolerations[]

Description

The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.

Type

object

Property	Type	Description
effect	string	Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.
key	string	Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is empty, operator must be Exists; this combination means to match all values and all keys.
operator	string	Operator represents a key's relationship to the value. Valid operators are Exists and Equal. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category.
tolerationSeconds	integer	TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is not set, which means tolerate the taint

Property	Type	Description
		forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.
<code>value</code>	<code>string</code>	Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

.spec.mgr.securityContext

Description

SecurityContext holds Pod-level security attributes and common Container settings.

Type

`object`

Property	Type	Description
<code>appArmorProfile</code>	<code>object</code>	appArmorProfile is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.
<code>fsGroup</code>	<code>integer</code>	A special supplemental group that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod: 1. The owning GID will be the FSGroup 2. The setgid bit is set (new files created in the volume will be owned by FSGroup)

Property	Type	Description
		3. The permission bits are OR'd with rw-rw-- -- If unset, the Kubelet will not modify the ownership and permissions of any volume. Note that this field cannot be set when spec.os.name is windows.
fsGroupChangePolicy	string	fsGroupChangePolicy defines behavior of changing ownership and permission of the volume before being exposed inside Pod. This field will only apply to volume types which support fsGroup based ownership(and permissions). It will have no effect on ephemeral volume types such as: secret, configmaps and emptydir. Valid values are "OnRootMismatch" and "Always". If not specified, "Always" is used. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
<code>runAsNonRoot</code>	<code>boolean</code>	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
<code>runAsUser</code>	<code>integer</code>	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
<code>seLinuxChangePolicy</code>	<code>string</code>	<code>seLinuxChangePolicy</code> defines how the container's SELinux label is applied to all volumes used by the Pod. It has no effect on nodes that do not support SELinux or to volumes does not support SELinux. Valid values are "MountOption" and "Recursive". "Recursive" means relabeling of all files on all Pod volumes by the container runtime. This may be slow for large volumes, but allows mixing privileged and unprivileged

Property	Type	Description
		Pods sharing the same volume on the same node. "MountOption" mounts all eligible Pod volumes with <code>-o context</code> mount option. This requires all Pods that share the same volume to use the same SELinux label. It is not possible to share the same volume among privileged and unprivileged Pods. Eligible volumes are in-tree FibreChannel and iSCSI volumes, and all CSI volumes whose CSI driver announces SELinux support by setting <code>spec.seLinuxMount: true</code> in their CSIDriver instance. Other volumes are always re-labelled recursively. "MountOption" value is allowed only when SELinuxMount feature gate is enabled. If not specified and SELinuxMount feature gate is enabled, "MountOption" is used. If not specified and SELinuxMount feature gate is disabled, "MountOption" is used for <code>ReadWriteOncePod</code> volumes and "Recursive" for all other volumes. This field affects only Pods that have SELinux label set, either in <code>PodSecurityContext</code> or in <code>SecurityContext</code> of all containers. All Pods that use the same volume should use the same <code>seLinuxChangePolicy</code>, otherwise some pods can get stuck in <code>ContainerCreating</code> state. Note that this field cannot be set when <code>spec.os.name</code> is <code>windows</code>.

Property	Type	Description
seLinuxOptions	object	The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
seccompProfile	object	The seccomp options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
<code>supplementalGroups</code>	<code>array</code>	A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.
<code>supplementalGroupsPolicy</code>	<code>string</code>	Defines how supplemental groups of the first container processes are calculated. Valid values are "Merge" and "Strict". If not specified, "Merge" is used. (Alpha) Using the field requires the SupplementalGroupsPolicy feature gate to be enabled and the container runtime must implement support for this feature. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
<code>sysctls</code>	<code>array</code>	Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.
<code>windowsOptions</code>	<code>object</code>	The Windows specific settings applied to all containers. If unspecified, the options within a container's SecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

`.spec.mgr.securityContext.appArmorProfile`

Description

appArmorProfile is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured

Property	Type	Description
		on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
<code>type</code>	<code>string</code>	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

.spec.mgr.securityContext.seLinuxOptions

Description

The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Property	Type	Description
<code>level</code>	<code>string</code>	Level is SELinux level label that applies to the container.
<code>role</code>	<code>string</code>	Role is a SELinux role label that applies to the container.
<code>type</code>	<code>string</code>	Type is a SELinux type label that applies to the container.

Property	Type	Description
<code>user</code>	<code>string</code>	User is a SELinux user label that applies to the container.

`.spec.mgr.securityContext.seccompProfile`

Description

The seccomp options to use by the containers in this pod. Note that this field cannot be set when `spec.os.name` is windows.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.
<code>type</code>	<code>string</code>	<code>type</code> indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied.

.spec.mgr.securityContext.supplementalGroups

Description

A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.

Type

array

.spec.mgr.securityContext.supplementalGroups[]

Type

integer

.spec.mgr.securityContext.sysctls

Description

Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.

Type

array

.spec.mgr.securityContext.sysctls[]

Description

Sysctl defines a kernel parameter to be set

Type

object

Required

name

value

Property	Type	Description
name	string	Name of a property to set
value	string	Value of a property to set

.spec.mgr.securityContext.windowsOptions

Description

The Windows specific settings applied to all containers. If unspecified, the options within a container's SecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
gmsaCredentialSpec	string	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.

Property	Type	Description
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entryptoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

.spec.mgr.slowSQL

Description

SlowSQL define MySQL slow sql spec

Type

`object`

Property	Type	Description
<code>enabled</code>	<code>boolean</code>	Enabled means enable slow sql
<code>image</code>	<code>string</code>	Image means the image of the slow sql.
<code>resources</code>	<code>object</code>	Resources holds ResourceRequirements for the MySQL Agent & Server Containers.
<code>serverEndpoint</code>	<code>string</code>	ServerEndpoint means the server endpoint of the slow sql.

.spec.mgr.slowSQL.resources

Description

Resources holds ResourceRequirements for the MySQL Agent & Server Containers.

Type

`object`

Property	Type	Description
<code>claims</code>	<code>array</code>	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.

Property	Type	Description
<code>limits</code>	<code>object</code>	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗
<code>requests</code>	<code>object</code>	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

`.spec.mgr.slowSQL.resources.claims`

Description

Claims lists the names of resources, defined in `spec.resourceClaims`, that are used by this container. This is an alpha field and requires enabling the `DynamicResourceAllocation` feature gate. This field is immutable. It can only be set for containers.

Type

`array`

`.spec.mgr.slowSQL.resources.claims[]`

Description

ResourceClaim references one entry in `PodSpec.ResourceClaims`.

Type

`object`

Required

name

Property	Type	Description
name	string	Name must match the name of one entry in pod.spec.resourceClaims of the Pod where this field is used. It makes that resource available inside a container.
request	string	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

.spec.mgr.slowSQL.resources.limits

Description

Limits describes the maximum amount of compute resources allowed. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

.spec.mgr.slowSQL.resources.requests

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

.spec.mgr.sslSecret

Description

SSLSecret allows a user to specify custom CA certificate, server certificate and server key for group replication SSL.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

.spec.mgr.tolerations

Description

Tolerations allows specifying a list of tolerations for controlling which set of Nodes a Pod can be scheduled on

Type

array

.spec.mgr.tolerations[]

Description

The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.

Type

object

Property	Type	Description
<code>effect</code>	<code>string</code>	Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.
<code>key</code>	<code>string</code>	Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is empty, operator must be Exists; this combination means to match all values and all keys.
<code>operator</code>	<code>string</code>	Operator represents a key's relationship to the value. Valid operators are Exists and Equal. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category.
<code>tolerationSeconds</code>	<code>integer</code>	TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is not set, which means tolerate the taint forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.
<code>value</code>	<code>string</code>	Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

.spec.mgr.upgradeOption

Description

UpgradeOption defines the upgrade option for the MySQL cluster.

Type

object

Property	Type	Description
<code>autoUpgrade</code>	<code>boolean</code>	AutoUpgrade means the auto upgrade option
<code>crVersion</code>	<code>string</code>	CRVersion means the version of the CR

.spec.mgr.volumeClaimTemplate

Description

VolumeClaimTemplate allows a user to specify how volumes inside a MySQL cluster

Type

object

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources

Property	Type	Description
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
<code>metadata</code>	<code>ObjectMeta</code> ↗	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata ↗
<code>spec</code>	<code>object</code>	spec defines the desired characteristics of a volume requested by a pod author. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims ↗
<code>status</code>	<code>object</code>	status represents the current information/status of a persistent volume claim. Read-only. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims ↗

`.spec.mgr.volumeClaimTemplate.spec`

Description

spec defines the desired characteristics of a volume requested by a pod author. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims>

Type

object

Property	Type	Description
accessModes	array	accessModes contains the desired access modes volume should have. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1
dataSource	object	dataSource field can be used to specify either: • An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) • An existing PVC (PersistentVolumeClaim) If the provisioner or an external controller can support the specified data source, it will create a new volume based on the contents of the specified data source. When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If the namespace is specified, then dataSourceRef will not be copied to dataSource.
dataSourceRef	object	dataSourceRef specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a PersistentVolumeClaim object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replace the functionality of the dataSource field and as such

Property	Type	Description
		both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in dataSourceRef, dataSource isn't set to the same value and must be empty. There are three important differences between dataSource and dataSourceRef • While dataSource only allows two specific types of objects, dataSourceRef allows any non-core objects as well as PersistentVolumeClaim objects. • While dataSource ignores disallowed values (dropping them), dataSourceRef preserves all values, and generates an error if a disallowed value is specified. • While dataSource only allows local objects, dataSourceRef allows objects in any namespace (Beta) Using this field requires the AnyVolumeDataSource feature gate to be enabled (Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource feature gate to be enabled.
resources	object	resources represents the minimum resources the volume should have. If RecoverVolumeExpansionFailure feature is enabled users are allowed to specify resource requirement that are lower than previous value but must still be higher than capacity recorded in the status field of claim. More info:

Property	Type	Description
		https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources ↗
<code>selector</code>	<code>object</code>	selector is a label query over volumes to consider binding.
<code>storageClassName</code>	<code>string</code>	storageClassName is the name of the StorageClass required by the claim. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#class-1 ↗

Property	Type	Description
<code>volumeAttributesClassName</code>	<code>string</code>	<code>volumeAttributesClassName</code> may be used to set the <code>VolumeAttributesClass</code> used by this claim. If specified, the CSI driver will create or update the volume with attributes defined in the corresponding <code>VolumeAttributesClass</code>. This has a different purpose than <code>storageClassName</code>, it can be changed after the claim is created. An empty string value means that <code>VolumeAttributesClass</code> will be applied to the claim if it's not allowed to reset this field to empty string once it is set. If unspecified and the <code>PersistentVolumeClaim</code> is unbound, the default <code>VolumeAttributesClass</code> will be used by the <code>persistentvolume</code> controller if it exists. If the resource referred to by <code>volumeAttributesClass</code> does not exist, this <code>PersistentVolumeClaim</code> will be set to <code>Pending</code> state, as reflected by the <code>modifyVolumeStatus</code> field, until such as a resource exists. More info: https://kubernetes.io/docs/concepts/storage/volume-attributes-classes/ (Beta) Using this field requires the <code>VolumeAttributesClass</code> feature gate to be enabled (off by default).
<code>volumeMode</code>	<code>string</code>	<code>volumeMode</code> defines what type of volume is required by the claim. Value of <code>Filesystem</code> is implied when not included in claim spec.
<code>volumeName</code>	<code>string</code>	<code>volumeName</code> is the binding reference to the <code>PersistentVolume</code> backing this claim.

.spec.mgr.volumeClaimTemplate.spec.accessModes

Description

accessModes contains the desired access modes the volume should have. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1>

Type

array

.spec.mgr.volumeClaimTemplate.spec.accessModes[]

Type

string

.spec.mgr.volumeClaimTemplate.spec.dataSource

Description

dataSource field can be used to specify either:

- * An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot)
- * An existing PVC (PersistentVolumeClaim)

If the provisioner or an external controller can support the specified data source, it will create a new volume based on the contents of the specified data source. When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If the namespace is specified, then dataSourceRef will not be copied to dataSource.

Type

object

Required

kind

name

Property	Type	Description
apiGroup	string	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.

Property	Type	Description
<code>kind</code>	<code>string</code>	Kind is the type of resource being referenced
<code>name</code>	<code>string</code>	Name is the name of resource being referenced

.spec.mgr.volumeClaimTemplate.spec.dataSourceRef

Description

`dataSourceRef` specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a `PersistentVolumeClaim` object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replace the functionality of the `dataSource` field and as such if both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in `dataSourceRef`, both fields (`dataSource` and `dataSourceRef`) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in `dataSourceRef`, `dataSource` isn't set to the same value and must be empty. There are three important differences between `dataSource` and `dataSourceRef`:

- * While `dataSource` only allows two specific types of objects, `dataSourceRef` allows any non-core object, as well as `PersistentVolumeClaim` objects.
- * While `dataSource` ignores disallowed values (dropping them), `dataSourceRef` preserves all values, and generates an error if a disallowed value is specified.
- * While `dataSource` only allows local objects, `dataSourceRef` allows objects in any namespaces.

(Beta) Using this field requires the `AnyVolumeDataSource` feature gate to be enabled.
 (Alpha) Using the namespace field of `dataSourceRef` requires the `CrossNamespaceVolumeDataSource` feature gate to be enabled.

Type

`object`

Required

`kind`

`name`

Property	Type	Description
<code>apiGroup</code>	<code>string</code>	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
<code>kind</code>	<code>string</code>	Kind is the type of resource being referenced
<code>name</code>	<code>string</code>	Name is the name of resource being referenced
<code>namespace</code>	<code>string</code>	Namespace is the namespace of resource being referenced Note that when a namespace is specified, a gateway.networking.k8s.io/ReferenceGrant object is required in the referent namespace to allow that namespace's owner to accept the reference. See the ReferenceGrant documentation for details. (Alpha) This field requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

`.spec.mgr.volumeClaimTemplate.spec.resources`

Description

resources represents the minimum resources the volume should have. If RecoverVolumeExpansionFailure feature is enabled users are allowed to specify resource requirements that are lower than previous value but must still be higher than capacity recorded in the status field of the claim. More info:

<https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources>

Type

`object`

Property	Type	Description
<code>limits</code>	<code>object</code>	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗
<code>requests</code>	<code>object</code>	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

`.spec.mgr.volumeClaimTemplate.spec.resources.limits`

Description

Limits describes the maximum amount of compute resources allowed. More info:
<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.mgr.volumeClaimTemplate.spec.resources.requests`

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info:
<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

.spec.mgr.volumeClaimTemplate.spec.selector

Description

selector is a label query over volumes to consider for binding.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.mgr.volumeClaimTemplate.spec.selector.matchExpressions

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.mgr.volumeClaimTemplate.spec.selector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mgr.volumeClaimTemplate.spec.selector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.mgr.volumeClaimTemplate.spec.selector.matchExpressions[].values[]`

Type

`string`

`.spec.mgr.volumeClaimTemplate.spec.selector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

`object`

`.spec.mgr.volumeClaimTemplate.status`

Description

status represents the current information/status of a persistent volume claim. Read-only. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims>

Type

`object`

Property	Type	Description
<code>accessModes</code>	<code>array</code>	accessModes contains the actual access modes of the volume backing the PVC has. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1

Property	Type	Description
<code>allocatedResourceStatuses</code>	<code>object</code>	<code>allocatedResourceStatuses</code> stores status of the volume being resized for the given PVC. Key names must follow standard Kubernetes label syntax. Valid values are either: * Un-prefixed keys: - <code>storage</code> - the controller is resizing the volume. * Custom resources must use implementation-defined prefixed names such as <code>"example.com/my-custom-resource"</code>. Apart from these values - keys that are unprefixed or have <code>kubernetes.io</code> prefix are considered reserved and hence may not be used. <code>ClaimResourceStatus</code> can be in any of following states: - <code>ControllerResizeInProgress</code>: State set when resize controller starts resizing the volume. - <code>ControllerResizeFailed</code>: State set when resize controller has failed in resize controller with a termination error. - <code>NodeResizePending</code>: State set when resize controller has finished resizing the volume but further action is needed on the node. - <code>NodeResizeInProgress</code>: State set when kubelet is resizing the volume. - <code>NodeResizeFailed</code>: State set when resizing has failed in kubelet with a termination error. Transient errors don't set <code>NodeResizeFailed</code>. example: if expanding a PVC for more capacity, the <code>pvc.status.allocatedResourceStatus</code> field can be one of the following states: - <code>pvc.status.allocatedResourceStatus["storage"] = "ControllerResizeInProgress"</code> - - <code>pvc.status.allocatedResourceStatus["storage"] = "ControllerResizeFailed"</code> - - <code>pvc.status.allocatedResourceStatus["storage"] = "NodeResizePending"</code> - - <code>pvc.status.allocatedResourceStatus["storage"] = "NodeResizeInProgress"</code> - - <code>pvc.status.allocatedResourceStatus["storage"] = "NodeResizeFailed"</code> -

Property	Type	Description
		"NodeResizeFailed" When this field is not means that no resize operation is in progress given PVC. A controller that receives PVC update with unknown resourceName or ClaimResource should ignore the update for the purpose it is designed. For example - a controller that is responsible for resizing capacity of the volume ignore PVC updates that change other values associated with PVC. This is an alpha field and requires enabling RecoverVolumeExpansionFailure feature.
<code>allocatedResources</code>	<code>object</code>	<code>allocatedResources</code> tracks the resources allocated for a PVC including its capacity. Key names follow standard Kubernetes label syntax. Valid values are either: * Un-prefixed keys: - storage - the capacity of the volume. * Custom resources must use implementation-defined prefixed names such as "example.com/my-custom-resource" Apart from these values - keys that are unprefixed or have kubernetes.io prefix are considered reserved and hence may not be used. Capacity reported here may be larger than requested capacity when a volume expansion operation is requested. For storage quota, the larger value between <code>allocatedResources</code> and <code>PVC.spec.resources</code> is used. If <code>allocatedResources</code> is not set, <code>PVC.spec.resources</code> alone is used for quota calculation. If a volume expansion capacity request is lowered, <code>allocatedResources</code> is only lowered if there are no expansion operations in progress and if the

Property	Type	Description
		volume capacity is equal or lower than the capacity. A controller that receives PVC update with unknown resourceName should ignore the the purpose it was designed. For example controller that only is responsible for resizing of the volume, should ignore PVC updates change other valid resources associated v This is an alpha field and requires enablin RecoverVolumeExpansionFailure feature.
capacity	object	capacity represents the actual resources c underlying volume.
conditions	array	conditions is the current Condition of persi volume claim. If underlying persistent volu resized then the Condition will be set to 'R
currentVolumeAttributesClassName	string	currentVolumeAttributesClassName is the name of the VolumeAttributesClass the P\ When unset, there is no VolumeAttributeC to this PersistentVolumeClaim This is a be requires enabling VolumeAttributesClass f by default).
modifyVolumeStatus	object	ModifyVolumeStatus represents the status ControllerModifyVolume operation. When there is no ModifyVolume operation being

Property	Type	Description
		This is a beta field and requires enabling VolumeAttributesClass feature (off by default)
<code>phase</code>	<code>string</code>	phase represents the current phase of PersistentVolumeClaim.

`.spec.mgr.volumeClaimTemplate.status.accessModes`

Description

accessModes contains the actual access modes the volume backing the PVC has. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1>

Type

`array`

`.spec.mgr.volumeClaimTemplate.status.accessModes[]`

Type

`string`

`.spec.mgr.volumeClaimTemplate.status.allocatedResourceStatuses`

Description

allocatedResourceStatuses stores status of resource being resized for the given PVC. Key names follow standard Kubernetes label syntax. Valid values are either: * Un-prefixed keys: - storage - the capacity of the volume. * Custom resources must use implementation-defined prefixed names such as "example.com/my-custom-resource" Apart from above values - keys that are unprefixed or have kubernetes.io prefix are considered reserved and hence may not be used. ClaimResourceStatus can be in any of following states: - ControllerResizeInProgress: State set when resize controller starts resizing the volume in

control-plane. - ControllerResizeFailed: State set when resize has failed in resize controller with a terminal error. - NodeResizePending: State set when resize controller has finished resizing the volume but further resizing of volume is needed on the node. -

NodeResizeInProgress: State set when kubelet starts resizing the volume. -

NodeResizeFailed: State set when resizing has failed in kubelet with a terminal error.

Transient errors don't set NodeResizeFailed. For example: if expanding a PVC for more capacity - this field can be one of the following states: -

pvc.status.allocatedResourceStatus['storage'] = "ControllerResizeInProgress" -

pvc.status.allocatedResourceStatus['storage'] = "ControllerResizeFailed" -

pvc.status.allocatedResourceStatus['storage'] = "NodeResizePending" -

pvc.status.allocatedResourceStatus['storage'] = "NodeResizeInProgress" -

pvc.status.allocatedResourceStatus['storage'] = "NodeResizeFailed" When this field is not

set, it means that no resize operation is in progress for the given PVC. A controller that

receives PVC update with previously unknown resourceName or ClaimResourceStatus

should ignore the update for the purpose it was designed. For example - a controller that

only is responsible for resizing capacity of the volume, should ignore PVC updates that

change other valid resources associated with PVC. This is an alpha field and requires

enabling RecoverVolumeExpansionFailure feature.

Type

object

.spec.mgr.volumeClaimTemplate.status.allocatedResources

Description

allocatedResources tracks the resources allocated to a PVC including its capacity. Key names follow standard Kubernetes label syntax. Valid values are either: * Un-prefixed keys:

- storage - the capacity of the volume. * Custom resources must use implementation-

defined prefixed names such as "example.com/my-custom-resource" Apart from above

values - keys that are unprefixed or have kubernetes.io prefix are considered reserved and

hence may not be used. Capacity reported here may be larger than the actual capacity

when a volume expansion operation is requested. For storage quota, the larger value from

allocatedResources and PVC.spec.resources is used. If allocatedResources is not set,

PVC.spec.resources alone is used for quota calculation. If a volume expansion capacity

request is lowered, allocatedResources is only lowered if there are no expansion operations

in progress and if the actual volume capacity is equal or lower than the requested capacity.

A controller that receives PVC update with previously unknown resourceName should ignore the update for the purpose it was designed. For example - a controller that only is responsible for resizing capacity of the volume, should ignore PVC updates that change other valid resources associated with PVC. This is an alpha field and requires enabling RecoverVolumeExpansionFailure feature.

Type

object

.spec.mgr.volumeClaimTemplate.status.capacity

Description

capacity represents the actual resources of the underlying volume.

Type

object

.spec.mgr.volumeClaimTemplate.status.conditions

Description

conditions is the current Condition of persistent volume claim. If underlying persistent volume is being resized then the Condition will be set to 'Resizing'.

Type

array

.spec.mgr.volumeClaimTemplate.status.conditions[]

Description

PersistentVolumeClaimCondition contains details about state of pvc

Type

object

Required

status

type

Property	Type	Description
lastProbeTime	string	lastProbeTime is the time we probed the condition.
lastTransitionTime	string	lastTransitionTime is the time the condition transitioned from
message	string	message is the human-readable message indicating detail
reason	string	reason is a unique, this should be a short, machine understandable reason for condition's last transition. If it reports "Resizing" underlying persistent volume is being resized.
status	string	Status is the status of the condition. Can be True, False, Unknown. https://kubernetes.io/docs/reference/kubernetes-api/config-objects/persistent-volume-claim-conditions/#:~:text=state%20of%20pvc-,conditions.status,-(string)%20of%20the%20condition
type	string	Type is the type of the condition. More info: https://kubernetes.io/docs/reference/kubernetes-api/config-objects/persistent-volume-claim-conditions/#:~:text=set%20to%20%27ResizeStarted%27.-,PersistentVolumeClaim%20status%20contains%20details%20about%20the%20reason%20for%20the%20last%20transition

.spec.mgr.volumeClaimTemplate.status.modifyVolumeStatus

Description

ModifyVolumeStatus represents the status object of ControllerModifyVolume operation. When this is unset, there is no ModifyVolume operation being attempted. This is a beta field and requires enabling VolumeAttributesClass feature (off by default).

Type

object

Required

status

Property	Type	Description
status	string	status is the status of the ControllerModifyVolume operation. It can be in any of following states: • Pending Pending indicates that the PersistentVolumeClaim cannot be modified due to unmet requirements, such as the specified VolumeAttributesClass not existing. • InProgress InProgress indicates that the volume is being modified. • Infeasible Infeasible indicates that the request has been rejected as invalid by the CSI driver. To resolve the error, a valid VolumeAttributesClass needs to be specified. Note: New statuses can be added in the future. Consumers should check for unknown statuses and fail appropriately.

Property	Type	Description
<code>targetVolumeAttributesClassName</code>	<code>string</code>	<code>targetVolumeAttributesClassName</code> is the name of the <code>VolumeAttributesClass</code> the PVC currently being reconciled

`.spec.params`

Description

Params is the configuration of MySQL Server

Type

`object`

Property	Type	Description
<code>mysql</code>	<code>object</code>	
<code>router</code>	<code>object</code>	

`.spec.params.mysql`

Type

`object`

`.spec.params.router`

Type

`object`

`.spec.paras`

Description

Paras deprecated: this field is deprecated, please use spec.params instead

Type

object

.spec.pxc

Description

PXC use to configure PerconaXtraDBCluster

Type

object

Property	Type	Description
<code>allowUnsafeConfigurations</code>	boolean	
<code>backup</code>	object	
<code>clusterReplication</code>	object	
<code>crVersion</code>	string	
<code>dcDesc</code>	object	
<code>enableCRValidationWebhook</code>	boolean	
<code>haproxy</code>	object	
<code>initImage</code>	string	
<code>ipFamilyPrefer</code>	string	IPFamily represents the IP Family (IPv4 or IPv6). This type is used to express the family of an IP expressed by a type (e.g. service.spec.ipFamilies).

Property	Type	Description
logCollectorSecretName	string	
logcollector	object	
multiDcSpec	object	
mysqlDExporter	object	
pause	boolean	
platform	string	
proxysql	object	
pxc	object	
pxcLogPath	string	
restrictedPsaEnabled	boolean	
secretsName	string	
serviceMonitor	object	
slowsql	object	
sslInternalSecretName	string	
sslSecretName	string	
tls	object	
updateStrategy	string	StatefulSetUpdateStrategyType is a string enumeration type that enumerates all possible update strategies for the StatefulSet controller.

Property	Type	Description
<code>upgradeOptions</code>	<code>object</code>	
<code>vaultSecretName</code>	<code>string</code>	

`.spec.pxc.backup`

Type

`object`

Property	Type	Description
<code>annotations</code>	<code>object</code>	
<code>image</code>	<code>string</code>	
<code>imagePullPolicy</code>	<code>string</code>	PullPolicy describes a policy for if/when to pull a container image
<code>imagePullSecrets</code>	<code>array</code>	
<code>pitr</code>	<code>object</code>	
<code>schedule</code>	<code>array</code>	
<code>serviceAccountName</code>	<code>string</code>	
<code>storages</code>	<code>object</code>	

`.spec.pxc.backup.annotations`

Type

`object`

.spec.pxc.backup.imagePullSecrets

Type

array

.spec.pxc.backup.imagePullSecrets[]

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

.spec.pxc.backup.pitr

Type

object

Required

enabled

storageName

Property	Type	Description
enabled	boolean	

Property	Type	Description
resources	object	
storageName	string	
timeBetweenUploads	integer	

.spec.pxc.backup.pitr.resources

Type

object

Property	Type	Description
limits	object	
requests	object	

.spec.pxc.backup.pitr.resources.limits

Type

object

Property	Type	Description
cpu	string	
ephemeral-storage	string	
memory	string	

.spec.pxc.backup.pitr.resources.requests

Type

object

Property	Type	Description
cpu	string	
ephemeral-storage	string	
memory	string	

.spec.pxc.backup.schedule

Type

array

.spec.pxc.backup.schedule[]

Type

object

Property	Type	Description
keep	integer	
name	string	
schedule	string	
storageName	string	

.spec.pxc.backup.storages

Type

object

.spec.pxc.clusterReplication

Type

object

Property	Type	Description
enableSemiSync	boolean	
enabled	boolean	
isReplica	boolean	
nodePortEnabled	boolean	
peerEntryHost	string	
peerEntryPort	integer	

.spec.pxc.dcDesc

Type

object

Property	Type	Description
enableSemiSync	boolean	
extraProviderOptions	string	
id	string	
ipAddrs	array	
isReplica	boolean	
name	string	

Property	Type	Description
<code>nodePortEnabled</code>	<code>boolean</code>	
<code>nodeWeight</code>	<code>integer</code>	
<code>sourceDcList</code>	<code>array</code>	
<code>sstHosts</code>	<code>array</code>	
<code>sstPort</code>	<code>integer</code>	

`.spec.pxc.dcDesc.ipAddrs`

Type

`array`

`.spec.pxc.dcDesc.ipAddrs[]`

Type

`string`

`.spec.pxc.dcDesc.sourceDcList`

Type

`array`

`.spec.pxc.dcDesc.sourceDcList[]`

Type

`string`

`.spec.pxc.dcDesc.sstHosts`

Type

array

.spec.pxc.dcDesc.sstHosts[]**Type**

string

.spec.pxc.haproxy**Type**

object

Property	Type	Description
affinity	object	
annotations	object	
configuration	string	
containerSecurityContext	object	SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.
enableProbes	boolean	
enabled	boolean	

Property	Type	Description
<code>externalTrafficPolicy</code>	<code>string</code>	for backwards compat
<code>forceUnsafeBootstrap</code>	<code>boolean</code>	
<code>gracePeriod</code>	<code>integer</code>	
<code>image</code>	<code>string</code>	
<code>imagePullPolicy</code>	<code>string</code>	PullPolicy describes a policy for if/when to pull a container image
<code>imagePullSecrets</code>	<code>array</code>	
<code>labels</code>	<code>object</code>	
<code>livenessDelaySec</code>	<code>integer</code>	
<code>livenessProbes</code>	<code>object</code>	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
<code>loadBalancerSourceRanges</code>	<code>array</code>	
<code>nodeSelector</code>	<code>object</code>	
<code>podDisruptionBudget</code>	<code>object</code>	
<code>podSecurityContext</code>	<code>object</code>	PodSecurityContext holds pod-level security attributes and common container settings. Some fields are also present in

Property	Type	Description
		container.securityContext. Field values of container.securityContext take precedence over field values of PodSecurityContext.
priorityClassName	string	
readinessDelaySec	integer	
readinessProbes	object	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
replicasExternalTrafficPolicy	string	for backwards compat
replicasServiceType	string	Service Type string describes ingress methods for a service
resources	object	
schedulerName	string	
serverMaxConnections	integer	
serviceAccountName	string	
serviceAnnotations	object	

Property	Type	Description
<code>serviceType</code>	<code>string</code>	Service Type string describes ingress methods for a service
<code>sidecarResources</code>	<code>object</code>	
<code>size</code>	<code>integer</code>	
<code>sslInternalSecretName</code>	<code>string</code>	
<code>sslSecretName</code>	<code>string</code>	
<code>startupProbe</code>	<code>object</code>	
<code>tolerations</code>	<code>array</code>	
<code>vaultSecretName</code>	<code>string</code>	
<code>volumeSpec</code>	<code>object</code>	
<code>writeNode</code>	<code>string</code>	

`.spec.pxc.haproxy.affinity`

Type

`object`

Property	Type	Description
<code>advanced</code>	<code>object</code>	Affinity is a group of affinity scheduling rules.
<code>antiAffinityTopologyKey</code>	<code>string</code>	

.spec.pxc.haproxy.affinity.advanced

Description

Affinity is a group of affinity scheduling rules.

Type

object

Property	Type	Description
<code>nodeAffinity</code>	object	Describes node affinity scheduling rules for the pod.
<code>podAffinity</code>	object	Describes pod affinity scheduling rules (e.g. co-locate this pod in the same node, zone, etc. as some other pod(s)).
<code>podAntiAffinity</code>	object	Describes pod anti-affinity scheduling rules (e.g. avoid putting this pod in the same node, zone, etc. as some other pod(s)).

.spec.pxc.haproxy.affinity.advanced.nodeAffinity

Description

Describes node affinity scheduling rules for the pod.

Type

object

Property	Type	Description
<code>preferredDuringSchedulingIgnoredDuringExecution</code>	array	The scheduler will prefer to schedule pods to

Property	Type	Description
		nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution affinity expressions, etc.) compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>object</code>	If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some

Property	Type	Description
		point during pod execution (e.g. due to an update), the system may or may not try to eventually evict the pod from its node.

.spec.pxc.haproxy.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest sum are the most preferred.

Type

array

.spec.pxc.haproxy.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

Description

An empty preferred scheduling term matches all objects with implicit weight 0 (i.e. it's a no-op). A null preferred scheduling term matches no objects (i.e. is also a no-op).

Type

object

Required

preference

weight

Property	Type	Description
preference	object	A node selector term, associated with the corresponding weight.
weight	integer	Weight associated with matching the corresponding nodeSelectorTerm, in the range 1-100.

**.spec.pxc.haproxy.affinity.advanced.nodeAffinity.preferred
DuringSchedulingIgnoredDuringExecution[].preference**

Description

A node selector term, associated with the corresponding weight.

Type

object

Property	Type	Description
matchExpressions	array	A list of node selector requirements by node's labels.
matchFields	array	A list of node selector requirements by node's fields.

**.spec.pxc.haproxy.affinity.advanced.nodeAffinity.preferred
DuringSchedulingIgnoredDuringExecution[].preference.m
atchExpressions**

Description

A list of node selector requirements by node's labels.

Type

array

.spec.pxc.haproxy.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[]

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.

Property	Type	Description
<code>values</code>	<code>array</code>	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

`.spec.pxc.haproxy.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values`

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.pxc.haproxy.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values[]`

Type

`string`

`.spec.pxc.haproxy.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.m`

atchFields

Description

A list of node selector requirements by node's fields.

Type

array

.spec.pxc.haproxy.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matches[]

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will

Property	Type	Description
		be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.pxc.haproxy.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.pxc.haproxy.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values[]

Type

string

.spec.pxc.haproxy.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to

be met at some point during pod execution (e.g. due to an update), the system may or may not try to eventually evict the pod from its node.

Type

object

Required

nodeSelectorTerms

Property	Type	Description
nodeSelectorTerms	array	Required. A list of node selector terms. The terms are ORed.

.spec.pxc.haproxy.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms

Description

Required. A list of node selector terms. The terms are ORed.

Type

array

.spec.pxc.haproxy.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[]

Description

A null or empty node selector term matches no objects. The requirements of them are ANDed. The TopologySelectorTerm type implements a subset of the NodeSelectorTerm.

Type

object

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	A list of node selector requirements by node's labels.
<code>matchFields</code>	<code>array</code>	A list of node selector requirements by node's fields.

`.spec.pxc.haproxy.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions`

Description

A list of node selector requirements by node's labels.

Type

`array`

`.spec.pxc.haproxy.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[]`

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

`key`

`operator`

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist, Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.pxc.haproxy.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.pxc.haproxy.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorT

terms[].matchExpressions[].values[]

Type

string

.spec.pxc.haproxy.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields

Description

A list of node selector requirements by node's fields.

Type

array

.spec.pxc.haproxy.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[]

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	The label key that the selector applies to.

Property	Type	Description
<code>operator</code>	<code>string</code>	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
<code>values</code>	<code>array</code>	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

`.spec.pxc.haproxy.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values`

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.pxc.haproxy.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values[]`

Type

`string`

.spec.pxc.haproxy.affinity.advanced.podAffinity

Description

Describes pod affinity scheduling rules (e.g. co-locate this pod in the same node, zone, etc. as some other pod(s)).

Type

object

Property	Type	Description
<code>preferredDuringSchedulingIgnoredDuringExecution</code>	array	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution, affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the

Property	Type	Description
		node(s) with the highest sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	If the affinity requirement: specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirement: specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

`.spec.pxc.haproxy.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights,

i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

array

**.spec.pxc.haproxy.affinity.advanced.podAffinity.preferred
DuringSchedulingIgnoredDuringExecution[]**

Description

The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Type

object

Required

podAffinityTerm weight

Property	Type	Description
podAffinityTerm	object	Required. A pod affinity term, associated with the corresponding weight.
weight	integer	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

**.spec.pxc.haproxy.affinity.advanced.podAffinity.preferred
DuringSchedulingIgnoredDuringExecution[].podAffinityTe**

rm

Description

Required. A pod affinity term, associated with the corresponding weight.

Type

object

Required

topologyKey

Property	Type	Description
<code>labelSelector</code>	object	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	array	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>mismatchLabelKeys</code>	array	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The

Property	Type	Description
		keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both <code>mismatchLabelKeys</code> and <code>labelSelector</code>. Also, <code>mismatchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. This is a beta field and requires enabling <code>MatchLabelKeysInPodAffinity</code> feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the <code>namespaces</code> field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector (<code>{}</code>) matches all namespaces.
<code>namespaces</code>	<code>array</code>	<code>namespaces</code> specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by <code>namespaceSelector</code>. null or empty namespaces list and null <code>namespaceSelector</code> means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the <code>labelSelector</code> in the specified namespaces, where co-

Property	Type	Description
		located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

.spec.pxc.haproxy.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.pxc.haproxy.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTe

rm.labelSelector.matchExpressions

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.pxc.haproxy.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.

Property	Type	Description
<code>values</code>	<code>array</code>	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.pxc.haproxy.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.pxc.haproxy.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]`

Type

`string`

`.spec.pxc.haproxy.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.pxc.haproxy.affinity.advanced.podAffinity.preferred DuringSchedulingIgnoredDuringExecution[].podAffinityTe rm.matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.pxc.haproxy.affinity.advanced.podAffinity.preferred DuringSchedulingIgnoredDuringExecution[].podAffinityTe rm.matchLabelKeys[]

Type

string

.spec.pxc.haproxy.affinity.advanced.podAffinity.preferred DuringSchedulingIgnoredDuringExecution[].podAffinityTe

rm.mismatchLabelKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

**.spec.pxc.haproxy.affinity.advanced.podAffinity.preferred
DuringSchedulingIgnoredDuringExecution[].podAffinityTe
rm.mismatchLabelKeys[]**

Type

string

**.spec.pxc.haproxy.affinity.advanced.podAffinity.preferred
DuringSchedulingIgnoredDuringExecution[].podAffinityTe
rm.namespaceSelector**

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.pxc.haproxy.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.pxc.haproxy.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

**.spec.pxc.haproxy.affinity.advanced.podAffinity.preferred
DuringSchedulingIgnoredDuringExecution[].podAffinityTe
rm.namespaceSelector.matchExpressions[].values**

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

**.spec.pxc.haproxy.affinity.advanced.podAffinity.preferred
DuringSchedulingIgnoredDuringExecution[].podAffinityTe**

rm.namespaceSelector.matchExpressions[].values[]

Type

string

.spec.pxc.haproxy.affinity.advanced.podAffinity.preferred DuringSchedulingIgnoredDuringExecution[].podAffinityTe rm.namespaceSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.pxc.haproxy.affinity.advanced.podAffinity.preferred DuringSchedulingIgnoredDuringExecution[].podAffinityTe rm.namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.pxc.haproxy.affinity.advanced.podAffinity.preferred DuringSchedulingIgnoredDuringExecution[].podAffinityTe rm.namespaces[]

Type

string

.spec.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution**Description**

If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Type

array

.spec.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[]**Description**

Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running

Type

object

Required

topologyKey

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field

Property	Type	Description
		and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

.spec.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

keyoperator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]

Type

string

.spec.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also,

matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]

Type

string

.spec.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]

Type

string

.spec.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]

Type

string

.spec.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

`.spec.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces`

Description

`namespaces` specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by `namespaceSelector`. null or empty `namespaces` list and null `namespaceSelector` means "this pod's namespace".

Type

`array`

`.spec.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]`

Type

`string`

`.spec.pxc.haproxy.affinity.advanced.podAntiAffinity`

Description

Describes pod anti-affinity scheduling rules (e.g. avoid putting this pod in the same node, zone, etc. as some other pod(s)).

Type

`object`

Property	Type	Description
<code>preferredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but

Property	Type	Description
		it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution, anti-affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.
requiredDuringSchedulingIgnoredDuringExecution	array	If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod

Property	Type	Description
		execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling anti-affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

array

.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

Description

The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Type

object

Required

podAffinityTerm

weight

Property	Type	Description
podAffinityTerm	object	Required. A pod affinity term, associated with the corresponding weight.
weight	integer	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm

Description

Required. A pod affinity term, associated with the corresponding weight.

Type

object

Required

topologyKey

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field

Property	Type	Description
		and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinit

yTerm.labelSelector

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

object

Property	Type	Description
matchExpressions	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
matchLabels	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions

yTerm.labelSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This

array is replaced during a strategic merge patch.

Type

array

.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]

Type

string

.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti)

affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both `matchLabelKeys` and `labelSelector`. Also, `matchLabelKeys` cannot be set when `labelSelector` isn't set. This is a beta field and requires enabling `MatchLabelKeysInPodAffinity` feature gate (enabled by default).

Type

array

`.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]`

Type

string

`.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys`

Description

`MismatchLabelKeys` is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both `mismatchLabelKeys` and `labelSelector`. Also, `mismatchLabelKeys` cannot be set when `labelSelector` isn't set. This is a beta field and requires enabling `MatchLabelKeysInPodAffinity` feature gate (enabled by default).

Type

array

`.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys`

yTerm.mismatchLabelKeys[]

Type

string

.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
matchExpressions	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
matchLabels	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm

yTerm.namespaceSelector.matchExpressions

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.

Property	Type	Description
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]

Type

string

.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

string

.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the

system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Type

array

.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running

Type

object

Required

topologyKey

Property	Type	Description
labelSelector	object	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
matchLabelKeys	array	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity.

Property	Type	Description
		Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Property	Type	Description
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector

or.matchExpressions[].values[]

Type

string

.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

`.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]`

Type

string

`.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys`

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

`.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]`

Type

string

.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
matchExpressions	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
matchLabels	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespace

`eSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]`

Type

string

`.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

`.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces`

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.pxc.haproxy.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]

Type

string

.spec.pxc.haproxy.annotations

Type

object

.spec.pxc.haproxy.containerSecurityContext

Description

SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.

Type

object

Property	Type	Description
<code>allowPrivilegeEscalation</code>	<code>boolean</code>	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the <code>no_new_privs</code> flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: 1. run as Privileged 2. has CAP_SYS_ADMIN Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>appArmorProfile</code>	<code>object</code>	<code>appArmorProfile</code> is the AppArmor options to use by this container. If set, this profile overrides the pod's <code>appArmorProfile</code>. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>capabilities</code>	<code>object</code>	The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>privileged</code>	<code>boolean</code>	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when <code>spec.os.name</code> is windows.

Property	Type	Description
<code>procMount</code>	<code>string</code>	<code>procMount</code> denotes the type of proc mount to use for the containers. The default value is <code>Default</code> which uses the container runtime defaults for readonly paths and masked paths. This requires the <code>ProcMountType</code> feature flag to be enabled. Note that this field cannot be set when <code>spec.os.name</code> is <code>windows</code>.
<code>readOnlyRootFilesystem</code>	<code>boolean</code>	Whether this container has a read-only root filesystem. Default is <code>false</code>. Note that this field cannot be set when <code>spec.os.name</code> is <code>windows</code>.
<code>runAsGroup</code>	<code>integer</code>	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in <code>PodSecurityContext</code>. If set in both <code>SecurityContext</code> and <code>PodSecurityContext</code>, the value specified in <code>SecurityContext</code> takes precedence. Note that this field cannot be set when <code>spec.os.name</code> is <code>windows</code>.
<code>runAsNonRoot</code>	<code>boolean</code>	Indicates that the container must run as a non-root user. If <code>true</code>, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or <code>false</code>, no such validation will be performed. May also be set in <code>PodSecurityContext</code>. If set in

Property	Type	Description
		both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seccompProfile	object	The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
<code>windowsOptions</code>	<code>object</code>	The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

`.spec.pxc.haproxy.containerSecurityContext.appArmorProfile`

Description

`appArmorProfile` is the AppArmor options to use by this container. If set, this profile overrides the pod's `appArmorProfile`. Note that this field cannot be set when `spec.os.name` is `windows`.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if <code>type</code> is "Localhost".

Property	Type	Description
<code>type</code>	<code>string</code>	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

`.spec.pxc.haproxy.containerSecurityContext.capabilities`

Description

The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when `spec.os.name` is windows.

Type

`object`

Property	Type	Description
<code>add</code>	<code>array</code>	Added capabilities
<code>drop</code>	<code>array</code>	Removed capabilities

`.spec.pxc.haproxy.containerSecurityContext.capabilities.add`

Description

Added capabilities

Type

`array`

.spec.pxc.haproxy.containerSecurityContext.capabilities.add[]

Description

Capability represent POSIX capabilities type

Type

`string`

.spec.pxc.haproxy.containerSecurityContext.capabilities.dropped

Description

Removed capabilities

Type

`array`

.spec.pxc.haproxy.containerSecurityContext.capabilities.dropped[]

Description

Capability represent POSIX capabilities type

Type

`string`

.spec.pxc.haproxy.containerSecurityContext.seLinuxOptions

Description

The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Type

object

Property	Type	Description
level	string	Level is SELinux level label that applies to the container.
role	string	Role is a SELinux role label that applies to the container.
type	string	Type is a SELinux type label that applies to the container.
user	string	User is a SELinux user label that applies to the container.

.spec.pxc.haproxy.containerSecurityContext.seccompProfile

Description

The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.

Type

object

Required

type

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.
type	string	type indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied.

.spec.pxc.haproxy.containerSecurityContext.windowsoptions

Description

The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

.spec.pxc.haproxy.imagePullSecrets

Type

array

.spec.pxc.haproxy.imagePullSecrets[]

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

.spec.pxc.haproxy.labels

Type

object

.spec.pxc.haproxy.livenessProbes

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	Exec specifies a command to execute in the
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Default value is 3. Minimum value is 1.
grpc	object	GRPC specifies a GRPC HealthCheckRequest to
httpGet	object	HTTPGet specifies an HTTP GET request to
initialDelaySeconds	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes
periodSeconds	integer	How often (in seconds) to perform the probe. Default is 10 seconds. Minimum value is 1.
successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Default is 1. Must be 1 for liveness and startup. Minimum value is 1.

Property	Type	Description
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies a connection to a TCP
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod needs gracefully upon probe failure. The grace period duration in seconds after the processes running pod are sent a termination signal and the time processes are forcibly halted with a kill signal. Value longer than the expected cleanup time process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. spec.terminationGracePeriodSeconds is used.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/lifecycle#container-probes

`.spec.pxc.haproxy.livenessProbes.exec`

Description

Exec specifies a command to execute in the container.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.pxc.haproxy.livenessProbes.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

.spec.pxc.haproxy.livenessProbes.exec.command[]

Type

`string`

.spec.pxc.haproxy.livenessProbes.grpc

Description

GRPC specifies a GRPC HealthCheckRequest.

Type

`object`

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

.spec.pxc.haproxy.livenessProbes.httpGet

Description

HTTPGet specifies an HTTP GET request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.

Property	Type	Description
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

.spec.pxc.haproxy.livenessProbes.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

.spec.pxc.haproxy.livenessProbes.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name` `value`

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.pxc.haproxy.livenessProbes.tcpSocket`

Description

TCPSocket specifies a connection to a TCP port.

Type

`object`

Required

`port`

Property	Type	Description
<code>host</code>	<code>string</code>	Optional: Host name to connect to, defaults to the pod IP.
<code>port</code>		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.pxc.haproxy.loadBalancerSourceRanges`

Type

`array`

`.spec.pxc.haproxy.loadBalancerSourceRanges[]`

Type

`string`

`.spec.pxc.haproxy.nodeSelector`

Type

`object`

`.spec.pxc.haproxy.podDisruptionBudget`

Type

`object`

Property	Type	Description
<code>maxUnavailable</code>	<code>int</code>	Maximum number of pods that can be unavailable during the disruption.
<code>minAvailable</code>	<code>int</code>	Minimum number of pods that must remain available during the disruption.

`.spec.pxc.haproxy.podSecurityContext`

Description

PodSecurityContext holds pod-level security attributes and common container settings. Some fields are also present in container.securityContext. Field values of container.securityContext take precedence over field values of PodSecurityContext.

Type

`object`

Property	Type	Description
appArmorProfile	object	appArmorProfile is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.
fsGroup	integer	A special supplemental group that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod: 1. The owning GID will be the FSGroup 2. The setgid bit is set (new files created in the volume will be owned by FSGroup) 3. The permission bits are OR'd with rw-rw-- -- If unset, the Kubelet will not modify the ownership and permissions of any volume. Note that this field cannot be set when spec.os.name is windows.
fsGroupChangePolicy	string	fsGroupChangePolicy defines behavior of changing ownership and permission of the volume before being exposed inside Pod. This field will only apply to volume types which support fsGroup based ownership(and permissions). It will have no effect on ephemeral volume types such as: secret, configmaps and emptydir. Valid values are "OnRootMismatch" and "Always". If not specified, "Always" is used. Note that this

Property	Type	Description
		field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this

Property	Type	Description
		field cannot be set when spec.os.name is windows.
seLinuxChangePolicy	string	seLinuxChangePolicy defines how the container's SELinux label is applied to all volumes used by the Pod. It has no effect on nodes that do not support SELinux or to volumes does not support SELinux. Valid values are "MountOption" and "Recursive". "Recursive" means relabeling of all files on all Pod volumes by the container runtime. This may be slow for large volumes, but allows mixing privileged and unprivileged Pods sharing the same volume on the same node. "MountOption" mounts all eligible Pod volumes with <code>-o context</code> mount option. This requires all Pods that share the same volume to use the same SELinux label. It is not possible to share the same volume among privileged and unprivileged Pods. Eligible volumes are in-tree FibreChannel and iSCSI volumes, and all CSI volumes whose CSI driver announces SELinux support by setting spec.seLinuxMount: true in their CSIDriver instance. Other volumes are always re-labelled recursively. "MountOption" value is allowed only when SELinuxMount feature gate is enabled. If not specified and SELinuxMount feature gate is enabled, "MountOption" is used. If not specified and SELinuxMount feature

Property	Type	Description
		gate is disabled, "MountOption" is used for ReadWriteOncePod volumes and "Recursive" for all other volumes. This field affects only Pods that have SELinux label set, either in PodSecurityContext or in SecurityContext of all containers. All Pods that use the same volume should use the same seLinuxChangePolicy, otherwise some pods can get stuck in ContainerCreating state. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
seccompProfile	object	The seccomp options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
<code>supplementalGroups</code>	<code>array</code>	A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.
<code>supplementalGroupsPolicy</code>	<code>string</code>	Defines how supplemental groups of the first container processes are calculated. Valid values are "Merge" and "Strict". If not specified, "Merge" is used. (Alpha) Using the field requires the SupplementalGroupsPolicy feature gate to be enabled and the container runtime must implement support for this feature. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
<code>sysctls</code>	<code>array</code>	Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.
<code>windowsOptions</code>	<code>object</code>	The Windows specific settings applied to all containers. If unspecified, the options within a container's SecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

`.spec.pxc.haproxy.podSecurityContext.appArmorProfile`

Description

appArmorProfile is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured

Property	Type	Description
		on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
<code>type</code>	<code>string</code>	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

.spec.pxc.haproxy.podSecurityContext.seLinuxOptions

Description

The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Property	Type	Description
<code>level</code>	<code>string</code>	Level is SELinux level label that applies to the container.
<code>role</code>	<code>string</code>	Role is a SELinux role label that applies to the container.
<code>type</code>	<code>string</code>	Type is a SELinux type label that applies to the container.

Property	Type	Description
<code>user</code>	<code>string</code>	User is a SELinux user label that applies to the container.

`.spec.pxc.haproxy.podSecurityContext.seccompProfile`

Description

The seccomp options to use by the containers in this pod. Note that this field cannot be set when `spec.os.name` is windows.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.
<code>type</code>	<code>string</code>	<code>type</code> indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied.

`.spec.pxc.haproxy.podSecurityContext.supplementalGroups`

Description

A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.

Type

array

`.spec.pxc.haproxy.podSecurityContext.supplementalGroups[]`

Type

integer

`.spec.pxc.haproxy.podSecurityContext.sysctls`

Description

Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.

Type

array

`.spec.pxc.haproxy.podSecurityContext.sysctls[]`

Description

Sysctl defines a kernel parameter to be set

Type

object

Required

name

value

Property	Type	Description
name	string	Name of a property to set
value	string	Value of a property to set

.spec.pxc.haproxy.podSecurityContext.windowsOptions

Description

The Windows specific settings applied to all containers. If unspecified, the options within a container's SecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
gmsaCredentialSpec	string	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.

Property	Type	Description
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entryptoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

.spec.pxc.haproxy.readinessProbes

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

`object`

Property	Type	Description
<code>exec</code>	<code>object</code>	Exec specifies a command to execute in the
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe to be considered failed after having succeeded. Default value is 3. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies a GRPC HealthCheckRequest to
<code>httpGet</code>	<code>object</code>	HTTPGet specifies an HTTP GET request to
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe. Default is 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe to be considered successful after having failed. Default is 1. Must be 1 for liveness and startup. Minimum value is 1.
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies a connection to a TCP

Property	Type	Description
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod needs gracefully upon probe failure. The grace period duration in seconds after the processes running in the pod are sent a termination signal and the timer begins. Once the grace period has elapsed, the processes are forcibly halted with a kill signal. A value greater than zero indicates a grace period value longer than the expected cleanup time for the process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided in the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. <code>spec.terminationGracePeriodSeconds</code> is used.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/controllers/lifecycle#container-probes

`.spec.pxc.haproxy.readinessProbes.exec`

Description

Exec specifies a command to execute in the container.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.pxc.haproxy.readinessProbes.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

.spec.pxc.haproxy.readinessProbes.exec.command[]

Type

`string`

.spec.pxc.haproxy.readinessProbes.grpc

Description

GRPC specifies a GRPC HealthCheckRequest.

Type

`object`

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

.spec.pxc.haproxy.readinessProbes.httpGet

Description

HTTPGet specifies an HTTP GET request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.

Property	Type	Description
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.pxc.haproxy.readinessProbes.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.pxc.haproxy.readinessProbes.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name`

`value`

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.pxc.haproxy.readinessProbes.tcpSocket`

Description

TCPSocket specifies a connection to a TCP port.

Type

`object`

Required

`port`

Property	Type	Description
<code>host</code>	<code>string</code>	Optional: Host name to connect to, defaults to the pod IP.
<code>port</code>		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.pxc.haproxy.resources`

Type

`object`

Property	Type	Description
limits	object	
requests	object	

.spec.pxc.haproxy.resources.limits

Type

object

Property	Type	Description
cpu	string	
ephemeral-storage	string	
memory	string	

.spec.pxc.haproxy.resources.requests

Type

object

Property	Type	Description
cpu	string	
ephemeral-storage	string	
memory	string	

.spec.pxc.haproxy.serviceAnnotations

Type

`object`

`.spec.pxc.haproxy.sidecarResources`

Type

`object`

Property	Type	Description
<code>limits</code>	<code>object</code>	
<code>requests</code>	<code>object</code>	

`.spec.pxc.haproxy.sidecarResources.limits`

Type

`object`

Property	Type	Description
<code>cpu</code>	<code>string</code>	
<code>ephemeral-storage</code>	<code>string</code>	
<code>memory</code>	<code>string</code>	

`.spec.pxc.haproxy.sidecarResources.requests`

Type

`object`

Property	Type	Description
<code>cpu</code>	<code>string</code>	

Property	Type	Description
<code>ephemeral-storage</code>	<code>string</code>	
<code>memory</code>	<code>string</code>	

`.spec.pxc.haproxy.startupProbe`

Type

`object`

Property	Type	Description
<code>failureThreshold</code>	<code>integer</code>	
<code>initialDelaySeconds</code>	<code>integer</code>	
<code>periodSeconds</code>	<code>integer</code>	

`.spec.pxc.haproxy.tolerations`

Type

`array`

`.spec.pxc.haproxy.tolerations[]`

Description

The pod this Toleration is attached to tolerates any taint that matches the triple `<key,value,effect>` using the matching operator `<operator>`.

Type

`object`

Property	Type	Description
<code>effect</code>	<code>string</code>	Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.
<code>key</code>	<code>string</code>	Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is empty, operator must be Exists; this combination means to match all values and all keys.
<code>operator</code>	<code>string</code>	Operator represents a key's relationship to the value. Valid operators are Exists and Equal. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category.
<code>tolerationSeconds</code>	<code>integer</code>	TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is not set, which means tolerate the taint forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.
<code>value</code>	<code>string</code>	Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

.spec.pxc.haproxy.volumeSpec

Type

object

Property	Type	Description
<code>emptyDir</code>	object	EmptyDir to use as data volume for mysql. EmptyDir represents a temporary directory that shares a pod's lifetime.
<code>hostPath</code>	object	HostPath to use as data volume for mysql. HostPath represents a pre-existing file or directory on the host machine that is directly exposed to the container.
<code>persistentVolumeClaim</code>	object	PersistentVolumeClaim to specify PVC spec for the volume for mysql data. It has the highest level of precedence, followed by HostPath and EmptyDir. And represents the PVC specification.

.spec.pxc.haproxy.volumeSpec.emptyDir

Description

EmptyDir to use as data volume for mysql. EmptyDir represents a temporary directory that shares a pod's lifetime.

Type

object

Property	Type	Description
<code>medium</code>	<code>string</code>	medium represents what type of storage medium should back this directory. The default is "" which means to use the node's default medium. Must be an empty string (default) or Memory. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir
<code>sizeLimit</code>		sizeLimit is the total amount of local storage required for this EmptyDir volume. The size limit is also applicable for memory medium. The maximum usage on memory medium EmptyDir would be the minimum value between the SizeLimit specified here and the sum of memory limits of all containers in a pod. The default is nil which means that the limit is undefined. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir

`.spec.pxc.haproxy.volumeSpec.hostPath`

Description

HostPath to use as data volume for mysql. HostPath represents a pre-existing file or directory on the host machine that is directly exposed to the container.

Type

`object`

Required

`path`

Property	Type	Description
path	string	path of the directory on the host. If the path is a symlink, it will follow the link to the real path. More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath ↗
type	string	type for HostPath Volume Defaults to "" More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath ↗

.spec.pxc.haproxy.volumeSpec.persistentVolumeClaim

Description

PersistentVolumeClaim to specify PVC spec for the volume for mysql data. It has the highest level of precedence, followed by HostPath and EmptyDir. And represents the PVC specification.

Type

object

Property	Type	Description
accessModes	array	accessModes contains the desired access modes volume should have. More info: https://kubernetes.io/docs/concepts/storage/persistentvolumes#access-modes-1 ↗
dataSource	object	dataSource field can be used to specify either: An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot)

Property	Type	Description
		An existing PVC (PersistentVolumeClaim) If the provisioner or an external controller can support the specified data source, it will create a new volume based on the contents of the specified data source. When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If the namespace is specified, then dataSourceRef will not be copied to dataSource.
dataSourceRef	object	dataSourceRef specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a PersistentVolumeClaim object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will require the functionality of the dataSource field and as such both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in dataSourceRef, dataSource isn't set to the same value and must be empty. There are three important differences between dataSource and dataSourceRef While dataSource only allows two specific type objects, dataSourceRef allows any non-core object as well as PersistentVolumeClaim objects.

Property	Type	Description
		- While dataSource ignores disallowed values (dropping them), dataSourceRef preserves all values, and generates an error if a disallowed value is specified. - While dataSource only allows local objects, dataSourceRef allows objects in any namespace (Beta) Using this field requires the AnyVolumeDataSource feature gate to be enabled (Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource feature gate to be enabled.
resources	object	resources represents the minimum resources the volume should have. If RecoverVolumeExpansionFailure feature is enabled users are allowed to specify resource requirement that are lower than previous value but must still be higher than capacity recorded in the status field of claim. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources
selector	object	selector is a label query over volumes to consider binding.
storageClassName	string	storageClassName is the name of the StorageClass required by the claim. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#class-1

Property	Type	Description
<code>volumeAttributesClassName</code>	<code>string</code>	<code>volumeAttributesClassName</code> may be used to set the <code>VolumeAttributesClass</code> used by this claim. If specified, the CSI driver will create or update the volume with attributes defined in the corresponding <code>VolumeAttributesClass</code>. This has a different purpose than <code>storageClassName</code>, it can be changed after the claim is created. An empty string value means that <code>VolumeAttributesClass</code> will be applied to the claim if it's not allowed to reset this field to empty string once it is set. If unspecified and the <code>PersistentVolumeClaim</code> is unbound, the default <code>VolumeAttributesClass</code> will be used by the <code>persistentvolume</code> controller if it exists. If the resource referred to by <code>volumeAttributesClass</code> does not exist, this <code>PersistentVolumeClaim</code> will be set to <code>Pending</code> state, as reflected by the <code>modifyVolumeStatus</code> field, until such as a resource exists. More info: https://kubernetes.io/docs/concepts/storage/volume-attributes-classes/ (Beta) Using this field requires the <code>VolumeAttributesClass</code> feature gate to be enabled (off by default).
<code>volumeMode</code>	<code>string</code>	<code>volumeMode</code> defines what type of volume is required by the claim. Value of <code>Filesystem</code> is implied when not included in claim spec.
<code>volumeName</code>	<code>string</code>	<code>volumeName</code> is the binding reference to the <code>PersistentVolume</code> backing this claim.

`.spec.pxc.haproxy.volumeSpec.persistentVolumeClaim.accessModes`

Description

accessModes contains the desired access modes the volume should have. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1>

Type

array

`.spec.pxc.haproxy.volumeSpec.persistentVolumeClaim.accessModes[]`

Type

string

`.spec.pxc.haproxy.volumeSpec.persistentVolumeClaim.dataSource`

Description

dataSource field can be used to specify either: * An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) * An existing PVC (PersistentVolumeClaim) If the provisioner or an external controller can support the specified data source, it will create a new volume based on the contents of the specified data source. When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If the namespace is specified, then dataSourceRef will not be copied to dataSource.

Type

object

Required

kind

name

Property	Type	Description
<code>apiGroup</code>	<code>string</code>	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
<code>kind</code>	<code>string</code>	Kind is the type of resource being referenced
<code>name</code>	<code>string</code>	Name is the name of resource being referenced

`.spec.pxc.haproxy.volumeSpec.persistentVolumeClaim.dataSourceRef`

Description

`dataSourceRef` specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a `PersistentVolumeClaim` object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replace the functionality of the `dataSource` field and as such if both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in `dataSourceRef`, both fields (`dataSource` and `dataSourceRef`) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in `dataSourceRef`, `dataSource` isn't set to the same value and must be empty. There are three important differences between `dataSource` and `dataSourceRef`:

- * While `dataSource` only allows two specific types of objects, `dataSourceRef` allows any non-core object, as well as `PersistentVolumeClaim` objects.
- * While `dataSource` ignores disallowed values (dropping them), `dataSourceRef` preserves all values, and generates an error if a disallowed value is specified.
- * While `dataSource` only allows local objects, `dataSourceRef` allows objects in any namespaces.

(Beta) Using this field requires the `AnyVolumeDataSource` feature gate to be enabled.

(Alpha) Using the `namespace` field of `dataSourceRef` requires the `CrossNamespaceVolumeDataSource` feature gate to be enabled.

Type

object

Required

kind name

Property	Type	Description
apiGroup	string	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
kind	string	Kind is the type of resource being referenced
name	string	Name is the name of resource being referenced
namespace	string	Namespace is the namespace of resource being referenced Note that when a namespace is specified, a gateway.networking.k8s.io/ReferenceGrant object is required in the referent namespace to allow that namespace's owner to accept the reference. See the ReferenceGrant documentation for details. (Alpha) This field requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

.spec.pxc.haproxy.volumeSpec.persistentVolumeClaim.re
sources

Description

resources represents the minimum resources the volume should have. If RecoverVolumeExpansionFailure feature is enabled users are allowed to specify resource requirements that are lower than previous value but must still be higher than capacity recorded in the status field of the claim. More info:
<https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources>

Type

object

Property	Type	Description
<code>limits</code>	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
<code>requests</code>	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

`.spec.pxc.haproxy.volumeSpec.persistentVolumeClaim.resources.limits`

Description

Limits describes the maximum amount of compute resources allowed. More info:
<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

.spec.pxc.haproxy.volumeSpec.persistentVolumeClaim.requests

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info:

<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

.spec.pxc.haproxy.volumeSpec.persistentVolumeClaim.selector

Description

selector is a label query over volumes to consider for binding.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.pxc.haproxy.volumeSpec.persistentVolumeClaim.selector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.pxc.haproxy.volumeSpec.persistentVolumeClaim.selector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
<code>key</code>	<code>string</code>	key is the label key that the selector applies to.
<code>operator</code>	<code>string</code>	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.

Property	Type	Description
<code>values</code>	<code>array</code>	<code>values</code> is an array of string values. If the operator is <code>In</code> or <code>NotIn</code> , the values array must be non-empty. If the operator is <code>Exists</code> or <code>DoesNotExist</code> , the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.pxc.haproxy.volumeSpec.persistentVolumeClaim.selector.matchExpressions[].values`

Description

`values` is an array of string values. If the operator is `In` or `NotIn`, the values array must be non-empty. If the operator is `Exists` or `DoesNotExist`, the values array must be empty. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.pxc.haproxy.volumeSpec.persistentVolumeClaim.selector.matchExpressions[].values[]`

Type

`string`

`.spec.pxc.haproxy.volumeSpec.persistentVolumeClaim.selector.matchLabels`

Description

`matchLabels` is a map of {key,value} pairs. A single {key,value} in the `matchLabels` map is equivalent to an element of `matchExpressions`, whose key field is "key", the operator is `In`, and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.pxc.logcollector

Type

object

Property	Type	Description
configuration	string	
containerSecurityContext	object	SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.
enabled	boolean	
image	string	
imagePullPolicy	string	PullPolicy describes a policy for if/when to pull a container image
resources	object	

.spec.pxc.logcollector.containerSecurityContext

Description

SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.

Type

object

Property	Type	Description
<code>allowPrivilegeEscalation</code>	<code>boolean</code>	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the <code>no_new_privs</code> flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: 1. run as Privileged 2. has CAP_SYS_ADMIN Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>appArmorProfile</code>	<code>object</code>	<code>appArmorProfile</code> is the AppArmor options to use by this container. If set, this profile overrides the pod's <code>appArmorProfile</code>. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>capabilities</code>	<code>object</code>	The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>privileged</code>	<code>boolean</code>	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to

Property	Type	Description
		false. Note that this field cannot be set when spec.os.name is windows.
procMount	string	procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows.
readOnlyRootFilesystem	boolean	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to

Property	Type	Description
		start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seccompProfile	object	The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that

Property	Type	Description
		this field cannot be set when spec.os.name is windows.
windowsOptions	object	The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

.spec.pxc.logcollector.containerSecurityContext.appArmorProfile

Description

appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when spec.os.name is windows.

Type

object

Required

type

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
<code>type</code>	<code>string</code>	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

`.spec.pxc.logcollector.containerSecurityContext.capabilities`

Description

The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when `spec.os.name` is windows.

Type

`object`

Property	Type	Description
<code>add</code>	<code>array</code>	Added capabilities
<code>drop</code>	<code>array</code>	Removed capabilities

.spec.pxc.logcollector.containerSecurityContext.capabilities.add

Description

Added capabilities

Type

array

.spec.pxc.logcollector.containerSecurityContext.capabilities.add[]

Description

Capability represent POSIX capabilities type

Type

string

.spec.pxc.logcollector.containerSecurityContext.capabilities.drop

Description

Removed capabilities

Type

array

.spec.pxc.logcollector.containerSecurityContext.capabilities.drop[]

Description

Capability represent POSIX capabilities type

Type

string

.spec.pxc.logcollector.containerSecurityContext.seLinuxOptions

Description

The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Type

object

Property	Type	Description
level	string	Level is SELinux level label that applies to the container.
role	string	Role is a SELinux role label that applies to the container.
type	string	Type is a SELinux type label that applies to the container.
user	string	User is a SELinux user label that applies to the container.

.spec.pxc.logcollector.containerSecurityContext.seccompProfile

Description

The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.

Type

object

Required

type

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.
type	string	type indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied.

.spec.pxc.logcollector.containerSecurityContext.windows Options

Description

The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the

value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the

Property	Type	Description
		value specified in SecurityContext takes precedence.

.spec.pxc.logcollector.resources

Type

object

Property	Type	Description
limits	object	
requests	object	

.spec.pxc.logcollector.resources.limits

Type

object

Property	Type	Description
cpu	string	
ephemeral-storage	string	
memory	string	

.spec.pxc.logcollector.resources.requests

Type

object

Property	Type	Description
<code>cpu</code>	<code>string</code>	
<code>ephemeral-storage</code>	<code>string</code>	
<code>memory</code>	<code>string</code>	

`.spec.pxc.multiDcSpec`

Type

`object`

Property	Type	Description
<code>dcMembers</code>	<code>array</code>	
<code>enabled</code>	<code>boolean</code>	
<code>myId</code>	<code>integer</code>	

`.spec.pxc.multiDcSpec.dcMembers`

Type

`array`

`.spec.pxc.multiDcSpec.dcMembers[]`

Type

`object`

Property	Type	Description
<code>enableSemiSync</code>	<code>boolean</code>	

Property	Type	Description
<code>extraProviderOptions</code>	<code>string</code>	
<code>id</code>	<code>string</code>	
<code>ipAdrrs</code>	<code>array</code>	
<code>isReplica</code>	<code>boolean</code>	
<code>name</code>	<code>string</code>	
<code>nodePortEnabled</code>	<code>boolean</code>	
<code>nodeWeight</code>	<code>integer</code>	
<code>sourceDcList</code>	<code>array</code>	
<code>sstHosts</code>	<code>array</code>	
<code>sstPort</code>	<code>integer</code>	

`.spec.pxc.multiDcSpec.dcMembers[].ipAdrrs`

Type

`array`

`.spec.pxc.multiDcSpec.dcMembers[].ipAdrrs[]`

Type

`string`

`.spec.pxc.multiDcSpec.dcMembers[].sourceDcList`

Type

`array`

`.spec.pxc.multiDcSpec.dcMembers[].sourceDcList[]`

Type

string

`.spec.pxc.multiDcSpec.dcMembers[].sstHosts`

Type

array

`.spec.pxc.multiDcSpec.dcMembers[].sstHosts[]`

Type

string

`.spec.pxc.mysqlExporter`

Type

object

Property	Type	Description
<code>enabled</code>	boolean	
<code>image</code>	string	
<code>resources</code>	object	

`.spec.pxc.mysqlExporter.resources`

Type

object

Property	Type	Description
limits	object	
requests	object	

`.spec.pxc.mysqlExporter.resources.limits`

Type

object

Property	Type	Description
cpu	string	
ephemeral-storage	string	
memory	string	

`.spec.pxc.mysqlExporter.resources.requests`

Type

object

Property	Type	Description
cpu	string	
ephemeral-storage	string	
memory	string	

`.spec.pxc.proxysql`

Type

object

Property	Type	Description
affinity	object	
annotations	object	
configuration	string	
containerSecurityContext	object	SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.
enableProbes	boolean	
enabled	boolean	
externalTrafficPolicy	string	for backwards compat
forceUnsafeBootstrap	boolean	
gracePeriod	integer	
image	string	
imagePullPolicy	string	PullPolicy describes a policy for if/when to pull a container image
imagePullSecrets	array	

Property	Type	Description
labels	object	
livenessDelaySec	integer	
livenessProbes	object	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
loadBalancerSourceRanges	array	
nodeSelector	object	
podDisruptionBudget	object	
podSecurityContext	object	PodSecurityContext holds pod-level security attributes and common container settings. Some fields are also present in container.securityContext. Field values of container.securityContext take precedence over field values of PodSecurityContext.
priorityClassName	string	
readinessDelaySec	integer	
readinessProbes	object	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Property	Type	Description
<code>replicasExternalTrafficPolicy</code>	<code>string</code>	for backwards compat
<code>replicasServiceType</code>	<code>string</code>	Service Type string describes ingress methods for a service
<code>resources</code>	<code>object</code>	
<code>schedulerName</code>	<code>string</code>	
<code>serverMaxConnections</code>	<code>integer</code>	
<code>serviceAccountName</code>	<code>string</code>	
<code>serviceAnnotations</code>	<code>object</code>	
<code>serviceType</code>	<code>string</code>	Service Type string describes ingress methods for a service
<code>sidecarResources</code>	<code>object</code>	
<code>size</code>	<code>integer</code>	
<code>sslInternalSecretName</code>	<code>string</code>	
<code>sslSecretName</code>	<code>string</code>	
<code>startupProbe</code>	<code>object</code>	
<code>tolerations</code>	<code>array</code>	
<code>vaultSecretName</code>	<code>string</code>	
<code>volumeSpec</code>	<code>object</code>	

Property	Type	Description
<code>writeNode</code>	<code>string</code>	

`.spec.pxc.proxysql.affinity`

Type

`object`

Property	Type	Description
<code>advanced</code>	<code>object</code>	Affinity is a group of affinity scheduling rules.
<code>antiAffinityTopologyKey</code>	<code>string</code>	

`.spec.pxc.proxysql.affinity.advanced`

Description

Affinity is a group of affinity scheduling rules.

Type

`object`

Property	Type	Description
<code>nodeAffinity</code>	<code>object</code>	Describes node affinity scheduling rules for the pod.
<code>podAffinity</code>	<code>object</code>	Describes pod affinity scheduling rules (e.g. co-locate this pod in the same node, zone, etc. as some other pod(s)).

Property	Type	Description
podAntiAffinity	object	Describes pod anti-affinity scheduling rules (e.g. avoid putting this pod in the same node, zone, etc. as some other pod(s)).

.spec.pxc.proxysql.affinity.advanced.nodeAffinity

Description

Describes node affinity scheduling rules for the pod.

Type

object

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution affinity expressions, etc.) compute a sum by

Property	Type	Description
		iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	object	If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to an update), the system may or may not try to eventually evict the pod from its node.

`.spec.pxc.proxysql.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights,

i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest sum are the most preferred.

Type

array

.spec.pxc.proxysql.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

Description

An empty preferred scheduling term matches all objects with implicit weight 0 (i.e. it's a no-op). A null preferred scheduling term matches no objects (i.e. is also a no-op).

Type

object

Required

preference weight

Property	Type	Description
preference	object	A node selector term, associated with the corresponding weight.
weight	integer	Weight associated with matching the corresponding nodeSelectorTerm, in the range 1-100.

.spec.pxc.proxysql.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference

Description

A node selector term, associated with the corresponding weight.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	A list of node selector requirements by node's labels.
<code>matchFields</code>	array	A list of node selector requirements by node's fields.

.spec.pxc.proxysql.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions

Description

A list of node selector requirements by node's labels.

Type

array

.spec.pxc.proxysql.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[]

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist, Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.pxc.proxysql.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.pxc.proxysql.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values[]

Type

string

.spec.pxc.proxysql.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields

Description

A list of node selector requirements by node's fields.

Type

array

.spec.pxc.proxysql.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[]

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist, Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.pxc.proxysql.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.pxc.proxysql.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.

matchFields[].values[]

Type

string

.spec.pxc.proxysql.affinity.advanced.nodeAffinity.required DuringSchedulingIgnoredDuringExecution

Description

If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to an update), the system may or may not try to eventually evict the pod from its node.

Type

object

Required

nodeSelectorTerms

Property	Type	Description
nodeSelectorTerms	array	Required. A list of node selector terms. The terms are ORed.

.spec.pxc.proxysql.affinity.advanced.nodeAffinity.required DuringSchedulingIgnoredDuringExecution.nodeSelectorT erms

Description

Required. A list of node selector terms. The terms are ORed.

Type

array

.spec.pxc.proxysql.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[]

Description

A null or empty node selector term matches no objects. The requirements of them are ANDed. The TopologySelectorTerm type implements a subset of the NodeSelectorTerm.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	A list of node selector requirements by node's labels.
<code>matchFields</code>	array	A list of node selector requirements by node's fields.

.spec.pxc.proxysql.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions

Description

A list of node selector requirements by node's labels.

Type

array

`.spec.pxc.proxysql.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[]`

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

`.spec.pxc.proxysql.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorT`

`terms[].matchExpressions[].values`

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

`.spec.pxc.proxysql.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values[]`

Type

string

`.spec.pxc.proxysql.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields`

Description

A list of node selector requirements by node's fields.

Type

array

`.spec.pxc.proxysql.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[]`

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

**.spec.pxc.proxysql.affinity.advanced.nodeAffinity.required
DuringSchedulingIgnoredDuringExecution.nodeSelectorT
erms[].matchFields[].values**

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

**.spec.pxc.proxysql.affinity.advanced.nodeAffinity.required
DuringSchedulingIgnoredDuringExecution.nodeSelectorT
erms[].matchFields[].values[]**

Type

string

.spec.pxc.proxysql.affinity.advanced.podAffinity

Description

Describes pod affinity scheduling rules (e.g. co-locate this pod in the same node, zone, etc. as some other pod(s)).

Type

object

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the

Property	Type	Description
		scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution, affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.
requiredDuringSchedulingIgnoredDuringExecution	array	If the affinity requirement specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirement specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are

Property	Type	Description
		intersected, i.e. all terms must be satisfied.

.spec.pxc.proxysql.affinity.advanced.podAffinity.preferred DuringSchedulingIgnoredDuringExecution

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

array

.spec.pxc.proxysql.affinity.advanced.podAffinity.preferred DuringSchedulingIgnoredDuringExecution[]

Description

The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Type

object

Required

podAffinityTerm

weight

Property	Type	Description
<code>podAffinityTerm</code>	<code>object</code>	Required. A pod affinity term, associated with the corresponding weight.
<code>weight</code>	<code>integer</code>	weight associated with matching the corresponding <code>podAffinityTerm</code> , in the range 1-100.

`.spec.pxc.proxysql.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm`

Description

Required. A pod affinity term, associated with the corresponding weight.

Type

`object`

Required

`topologyKey`

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	A label query over a set of resources, in this case pods. If it's null, this <code>PodAffinityTerm</code> matches with no Pods.
<code>matchLabelKeys</code>	<code>array</code>	<code>MatchLabelKeys</code> is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with

Property	Type	Description
		<code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both <code>matchLabelKeys</code> and <code>labelSelector</code> . Also, <code>matchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. This is a beta field and requires enabling <code>MatchLabelKeysInPodAffinity</code> feature gate (enabled by default).
<code>mismatchLabelKeys</code>	array	<code>MismatchLabelKeys</code> is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both <code>mismatchLabelKeys</code> and <code>labelSelector</code> . Also, <code>mismatchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. This is a beta field and requires enabling <code>MatchLabelKeysInPodAffinity</code> feature gate (enabled by default).
<code>namespaceSelector</code>	object	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the <code>namespaces</code> field. null selector and null or empty namespaces list means "this pod's

Property	Type	Description
		namespace". An empty selector ({} matches all namespaces.
namespaces	array	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
topologyKey	string	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

.spec.pxc.proxysql.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

object

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.pxc.proxysql.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.pxc.proxysql.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

**.spec.pxc.proxysql.affinity.advanced.podAffinity.preferred
DuringSchedulingIgnoredDuringExecution[].podAffinityTe
rm.labelSelector.matchExpressions[].values**

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

**.spec.pxc.proxysql.affinity.advanced.podAffinity.preferred
DuringSchedulingIgnoredDuringExecution[].podAffinityTe**

rm.labelSelector.matchExpressions[].values[]

Type

string

.spec.pxc.proxysql.affinity.advanced.podAffinity.preferred DuringSchedulingIgnoredDuringExecution[].podAffinityTe rm.labelSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.pxc.proxysql.affinity.advanced.podAffinity.preferred DuringSchedulingIgnoredDuringExecution[].podAffinityTe rm.matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.pxc.proxysql.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]

Type

string

.spec.pxc.proxysql.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.pxc.proxysql.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]

Type

string

`.spec.pxc.proxysql.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.pxc.proxysql.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.pxc.proxysql.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.pxc.proxysql.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTe

rm.namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

**.spec.pxc.proxysql.affinity.advanced.podAffinity.preferred
DuringSchedulingIgnoredDuringExecution[].podAffinityTe
rm.namespaceSelector.matchExpressions[].values[]**

Type

string

**.spec.pxc.proxysql.affinity.advanced.podAffinity.preferred
DuringSchedulingIgnoredDuringExecution[].podAffinityTe
rm.namespaceSelector.matchLabels**

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

**.spec.pxc.proxysql.affinity.advanced.podAffinity.preferred
DuringSchedulingIgnoredDuringExecution[].podAffinityTe
rm.namespaces**

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

**.spec.pxc.proxysql.affinity.advanced.podAffinity.preferred
DuringSchedulingIgnoredDuringExecution[].podAffinityTe
rm.namespaces[]**

Type

string

**.spec.pxc.proxysql.affinity.advanced.podAffinity.required
DuringSchedulingIgnoredDuringExecution**

Description

If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Type

array

**.spec.pxc.proxysql.affinity.advanced.podAffinity.required
DuringSchedulingIgnoredDuringExecution[]**

Description

Defines a set of pods (namely those matching the `labelSelector` relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key `<topologyKey>` matches that of any node on which a pod of the set of pods is running

Type

object

Required

topologyKey

Property	Type	Description
<code>labelSelector</code>	object	A label query over a set of resources, in this case pods. If it's null, this <code>PodAffinityTerm</code> matches with no Pods.
<code>matchLabelKeys</code>	array	<code>MatchLabelKeys</code> is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both <code>matchLabelKeys</code> and <code>labelSelector</code> . Also, <code>matchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. This is a beta field and requires enabling <code>MatchLabelKeysInPodAffinity</code> feature gate (enabled by default).
<code>mismatchLabelKeys</code>	array	<code>MismatchLabelKeys</code> is a set of pod label keys to select which pods will be taken into consideration. The

Property	Type	Description
		keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both <code>mismatchLabelKeys</code> and <code>labelSelector</code>. Also, <code>mismatchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. This is a beta field and requires enabling <code>MatchLabelKeysInPodAffinity</code> feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the <code>namespaces</code> field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector (<code>{}</code>) matches all namespaces.
<code>namespaces</code>	<code>array</code>	<code>namespaces</code> specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by <code>namespaceSelector</code>. null or empty namespaces list and null <code>namespaceSelector</code> means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the <code>labelSelector</code> in the specified namespaces, where co-

Property	Type	Description
		located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

.spec.pxc.proxysql.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.pxc.proxysql.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector. matchExpressions

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.pxc.proxysql.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.pxc.proxysql.affinity.advanced.podAffinity.required DuringSchedulingIgnoredDuringExecution[].labelSelector. matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.pxc.proxysql.affinity.advanced.podAffinity.required DuringSchedulingIgnoredDuringExecution[].labelSelector. matchExpressions[].values[]

Type

string

.spec.pxc.proxysql.affinity.advanced.podAffinity.required DuringSchedulingIgnoredDuringExecution[].labelSelector. matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

`.spec.pxc.proxysql.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys`

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

`.spec.pxc.proxysql.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]`

Type

string

`.spec.pxc.proxysql.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys`

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti)

affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both `mismatchLabelKeys` and `labelSelector`. Also, `mismatchLabelKeys` cannot be set when `labelSelector` isn't set. This is a beta field and requires enabling `MatchLabelKeysInPodAffinity` feature gate (enabled by default).

Type

array

`.spec.pxc.proxysql.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]`

Type

string

`.spec.pxc.proxysql.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the `namespaces` field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector (`{}`) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.

Property	Type	Description
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.pxc.proxysql.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.pxc.proxysql.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

`key`

`operator`

Property	Type	Description
<code>key</code>	<code>string</code>	key is the label key that the selector applies to.
<code>operator</code>	<code>string</code>	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
<code>values</code>	<code>array</code>	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.pxc.proxysql.affinity.advanced.podAffinity.required DuringSchedulingIgnoredDuringExecution[].namespaceS elector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

`array`

.spec.pxc.proxysql.affinity.advanced.podAffinity.required DuringSchedulingIgnoredDuringExecution[].namespaceS elector.matchExpressions[].values[]

Type

`string`

.spec.pxc.proxysql.affinity.advanced.podAffinity.required DuringSchedulingIgnoredDuringExecution[].namespaceS elector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

`object`

.spec.pxc.proxysql.affinity.advanced.podAffinity.required DuringSchedulingIgnoredDuringExecution[].namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

`array`

.spec.pxc.proxysql.affinity.advanced.podAffinity.required DuringSchedulingIgnoredDuringExecution[].namespaces[]

Type

`string`

.spec.pxc.proxysql.affinity.advanced.podAntiAffinity

Description

Describes pod anti-affinity scheduling rules (e.g. avoid putting this pod in the same node, zone, etc. as some other pod(s)).

Type

object

Property	Type	Description
<code>preferredDuringSchedulingIgnoredDuringExecution</code>	array	The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, <code>requiredDuringSchedulingIgnoredDuringExecution</code> , anti-affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding <code>podAffinityTerm</code> ; the

Property	Type	Description
		node(s) with the highest sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

`.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the

expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling anti-affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

array

.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

Description

The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Type

object

Required

podAffinityTerm

weight

Property	Type	Description
<code>podAffinityTerm</code>	<code>object</code>	Required. A pod affinity term, associated with the corresponding weight.
<code>weight</code>	<code>integer</code>	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffini

tyTerm

Description

Required. A pod affinity term, associated with the corresponding weight.

Type

object

Required

topologyKey

Property	Type	Description
labelSelector	object	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
matchLabelKeys	array	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
mismatchLabelKeys	array	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The

Property	Type	Description
		keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both <code>mismatchLabelKeys</code> and <code>labelSelector</code>. Also, <code>mismatchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. This is a beta field and requires enabling <code>MatchLabelKeysInPodAffinity</code> feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the <code>namespaces</code> field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector (<code>{}</code>) matches all namespaces.
<code>namespaces</code>	<code>array</code>	<code>namespaces</code> specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by <code>namespaceSelector</code>. null or empty namespaces list and null <code>namespaceSelector</code> means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the <code>labelSelector</code> in the specified namespaces, where co-

Property	Type	Description
		located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffini`

tyTerm.labelSelector.matchExpressions

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.

Property	Type	Description
<code>values</code>	<code>array</code>	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]`

Type

`string`

`.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]

Type

string

.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffini

tyTerm.mismatchLabelKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]

Type

string

.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffini

tyTerm.namespaceSelector.matchExpressions[].values[]

Type

string

.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

string

.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Type

array

.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running

Type

object

Required

topologyKey

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field

Property	Type	Description
		and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelec`

tor

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

object

Property	Type	Description
matchExpressions	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
matchLabels	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector

tor.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This

array is replaced during a strategic merge patch.

Type

array

.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]

Type

string

.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti)

affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both `matchLabelKeys` and `labelSelector`. Also, `matchLabelKeys` cannot be set when `labelSelector` isn't set. This is a beta field and requires enabling `MatchLabelKeysInPodAffinity` feature gate (enabled by default).

Type

array

`.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]`

Type

string

`.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys`

Description

`MismatchLabelKeys` is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both `mismatchLabelKeys` and `labelSelector`. Also, `mismatchLabelKeys` cannot be set when `labelSelector` isn't set. This is a beta field and requires enabling `MatchLabelKeysInPodAffinity` feature gate (enabled by default).

Type

array

`.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatch`

LabelKeys[]

Type

string

.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespace

eSelector.matchExpressions

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.

Property	Type	Description
<code>values</code>	<code>array</code>	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]`

Type

`string`

`.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]

Type

string

.spec.pxc.proxysql.annotations

Type

object

.spec.pxc.proxysql.containerSecurityContext

Description

SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.

Type

object

Property	Type	Description
<code>allowPrivilegeEscalation</code>	<code>boolean</code>	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the <code>no_new_privs</code> flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: 1. run as Privileged 2. has CAP_SYS_ADMIN Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>appArmorProfile</code>	<code>object</code>	<code>appArmorProfile</code> is the AppArmor options to use by this container. If set, this profile overrides the pod's <code>appArmorProfile</code>. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>capabilities</code>	<code>object</code>	The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when <code>spec.os.name</code> is windows.

Property	Type	Description
<code>privileged</code>	<code>boolean</code>	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.
<code>procMount</code>	<code>string</code>	procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows.
<code>readOnlyRootFilesystem</code>	<code>boolean</code>	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
<code>runAsGroup</code>	<code>integer</code>	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
<code>runAsNonRoot</code>	<code>boolean</code>	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
<code>runAsUser</code>	<code>integer</code>	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
<code>seLinuxOptions</code>	<code>object</code>	The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
<code>seccompProfile</code>	<code>object</code>	The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.
<code>windowsOptions</code>	<code>object</code>	The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

`.spec.pxc.proxysql.containerSecurityContext.appArmorProfile`

Description

`appArmorProfile` is the AppArmor options to use by this container. If set, this profile overrides the pod's `appArmorProfile`. Note that this field cannot be set when `spec.os.name` is windows.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
<code>type</code>	<code>string</code>	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

`.spec.pxc.proxysql.containerSecurityContext.capabilities`

Description

The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Property	Type	Description
<code>add</code>	<code>array</code>	Added capabilities
<code>drop</code>	<code>array</code>	Removed capabilities

.spec.pxc.proxysql.containerSecurityContext.capabilities.add

Description

Added capabilities

Type

array

.spec.pxc.proxysql.containerSecurityContext.capabilities.add[]

Description

Capability represent POSIX capabilities type

Type

string

.spec.pxc.proxysql.containerSecurityContext.capabilities.drop

Description

Removed capabilities

Type

array

.spec.pxc.proxysql.containerSecurityContext.capabilities.drop[]

Description

Capability represent POSIX capabilities type

Type

string

`.spec.pxc.proxysql.containerSecurityContext.seLinuxOptions`

Description

The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Type

object

Property	Type	Description
<code>level</code>	string	Level is SELinux level label that applies to the container.
<code>role</code>	string	Role is a SELinux role label that applies to the container.
<code>type</code>	string	Type is a SELinux type label that applies to the container.
<code>user</code>	string	User is a SELinux user label that applies to the container.

`.spec.pxc.proxysql.containerSecurityContext.seccompProfile`

Description

The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.

Type

object

Required

type

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.
type	string	type indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied.

.spec.pxc.proxysql.containerSecurityContext.windowsOptions

Description

The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the

value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the

Property	Type	Description
		value specified in SecurityContext takes precedence.

`.spec.pxc.proxysql.imagePullSecrets`

Type

array

`.spec.pxc.proxysql.imagePullSecrets[]`

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

`.spec.pxc.proxysql.labels`

Type

object

.spec.pxc.proxysql.livenessProbes

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
<code>exec</code>	object	Exec specifies a command to execute in the
<code>failureThreshold</code>	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Default value is 3. Minimum value is 1.
<code>grpc</code>	object	GRPC specifies a GRPC HealthCheckRequest to
<code>httpGet</code>	object	HTTPGet specifies an HTTP GET request to
<code>initialDelaySeconds</code>	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes
<code>periodSeconds</code>	integer	How often (in seconds) to perform the probe. Default is 10 seconds. Minimum value is 1.

Property	Type	Description
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies a connection to a TCP endpoint.
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod needs to gracefully shutdown before it is terminated. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. A value greater than 0 indicates the grace period should be used. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. spec.terminationGracePeriodSeconds is used if specified.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More information: https://kubernetes.io/docs/concepts/workloads/pods/probe-lifecycle#container-probes

.spec.pxc.proxysql.livenessProbes.exec

Description

Exec specifies a command to execute in the container.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.pxc.proxysql.livenessProbes.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

.spec.pxc.proxysql.livenessProbes.exec.command[]

Type

string

.spec.pxc.proxysql.livenessProbes.grpc

Description

GRPC specifies a GRPC HealthCheckRequest.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

.spec.pxc.proxysql.livenessProbes.httpGet

Description

HTTPGet specifies an HTTP GET request to perform.

Type

object

Required

port

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.pxc.proxysql.livenessProbes.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.pxc.proxysql.livenessProbes.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

namevalue

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.pxc.proxysql.livenessProbes.tcpSocket

Description

TCPSocket specifies a connection to a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

.spec.pxc.proxysql.loadBalancerSourceRanges

Type

array

.spec.pxc.proxysql.loadBalancerSourceRanges[]

Type

string

.spec.pxc.proxysql.nodeSelector

Type

object

.spec.pxc.proxysql.podDisruptionBudget

Type

object

Property	Type	Description
maxUnavailable		
minAvailable		

.spec.pxc.proxysql.podSecurityContext

Description

PodSecurityContext holds pod-level security attributes and common container settings. Some fields are also present in container.securityContext. Field values of container.securityContext take precedence over field values of PodSecurityContext.

Type

object

Property	Type	Description
appArmorProfile	object	appArmorProfile is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.
fsGroup	integer	A special supplemental group that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod: 1. The owning GID will be the FSGroup 2. The setgid bit is set (new files created in the volume will be owned by FSGroup) 3. The permission bits are OR'd with rw-rw-- -- If unset, the Kubelet will not modify the ownership and permissions of any volume. Note that this field cannot be set when spec.os.name is windows.
fsGroupChangePolicy	string	fsGroupChangePolicy defines behavior of changing ownership and permission of the volume before being exposed inside Pod. This field will only apply to volume types which support fsGroup based ownership(and permissions). It will have no effect on ephemeral volume types such as: secret, configmaps and emptydir. Valid values are "OnRootMismatch" and "Always". If not

Property	Type	Description
		specified, "Always" is used. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this

Property	Type	Description
		field cannot be set when spec.os.name is windows.
seLinuxChangePolicy	string	seLinuxChangePolicy defines how the container's SELinux label is applied to all volumes used by the Pod. It has no effect on nodes that do not support SELinux or to volumes does not support SELinux. Valid values are "MountOption" and "Recursive". "Recursive" means relabeling of all files on all Pod volumes by the container runtime. This may be slow for large volumes, but allows mixing privileged and unprivileged Pods sharing the same volume on the same node. "MountOption" mounts all eligible Pod volumes with <code>-o context</code> mount option. This requires all Pods that share the same volume to use the same SELinux label. It is not possible to share the same volume among privileged and unprivileged Pods. Eligible volumes are in-tree FibreChannel and iSCSI volumes, and all CSI volumes whose CSI driver announces SELinux support by setting spec.seLinuxMount: true in their CSIDriver instance. Other volumes are always re-labelled recursively. "MountOption" value is allowed only when SELinuxMount feature gate is enabled. If not specified and SELinuxMount feature gate is enabled, "MountOption" is used. If not specified and SELinuxMount feature

Property	Type	Description
		gate is disabled, "MountOption" is used for ReadWriteOncePod volumes and "Recursive" for all other volumes. This field affects only Pods that have SELinux label set, either in PodSecurityContext or in SecurityContext of all containers. All Pods that use the same volume should use the same seLinuxChangePolicy, otherwise some pods can get stuck in ContainerCreating state. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
seccompProfile	object	The seccomp options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
<code>supplementalGroups</code>	<code>array</code>	A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.
<code>supplementalGroupsPolicy</code>	<code>string</code>	Defines how supplemental groups of the first container processes are calculated. Valid values are "Merge" and "Strict". If not specified, "Merge" is used. (Alpha) Using the field requires the SupplementalGroupsPolicy feature gate to be enabled and the container runtime must implement support for this feature. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
<code>sysctls</code>	<code>array</code>	Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.
<code>windowsOptions</code>	<code>object</code>	The Windows specific settings applied to all containers. If unspecified, the options within a container's SecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

`.spec.pxc.proxysql.podSecurityContext.appArmorProfile`

Description

appArmorProfile is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured

Property	Type	Description
		on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
<code>type</code>	<code>string</code>	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

`.spec.pxc.proxysql.podSecurityContext.seLinuxOptions`

Description

The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when `spec.os.name` is windows.

Type

`object`

Property	Type	Description
<code>level</code>	<code>string</code>	Level is SELinux level label that applies to the container.
<code>role</code>	<code>string</code>	Role is a SELinux role label that applies to the container.
<code>type</code>	<code>string</code>	Type is a SELinux type label that applies to the container.

Property	Type	Description
<code>user</code>	<code>string</code>	User is a SELinux user label that applies to the container.

`.spec.pxc.proxysql.podSecurityContext.seccompProfile`

Description

The seccomp options to use by the containers in this pod. Note that this field cannot be set when `spec.os.name` is windows.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.
<code>type</code>	<code>string</code>	<code>type</code> indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied.

.spec.pxc.proxysql.podSecurityContext.supplementalGroups

Description

A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.

Type

array

.spec.pxc.proxysql.podSecurityContext.supplementalGroups[]

Type

integer

.spec.pxc.proxysql.podSecurityContext.sysctls

Description

Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.

Type

array

.spec.pxc.proxysql.podSecurityContext.sysctls[]

Description

Sysctl defines a kernel parameter to be set

Type

object

Required

name

value

Property	Type	Description
name	string	Name of a property to set
value	string	Value of a property to set

.spec.pxc.proxysql.podSecurityContext.windowsOptions**Description**

The Windows specific settings applied to all containers. If unspecified, the options within a container's SecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
gmsaCredentialSpec	string	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the GMSACredentialSpecName field.

Property	Type	Description
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

`.spec.pxc.proxysql.readinessProbes`

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

`object`

Property	Type	Description
<code>exec</code>	<code>object</code>	Exec specifies a command to execute in the
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe to be considered failed after having succeeded. Default value is 3. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies a GRPC HealthCheckRequest to the service.
<code>httpGet</code>	<code>object</code>	HTTPGet specifies an HTTP GET request to the container's port.
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe to be considered successful after having failed. Default is 1. Must be 1 for liveness and startup. Minimum value is 1.
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies a connection to a TCP port.

Property	Type	Description
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod needs gracefully upon probe failure. The grace period duration in seconds after the processes running pod are sent a termination signal and the time processes are forcibly halted with a kill signal. A value longer than the expected cleanup time process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. <code>spec.terminationGracePeriodSeconds</code> is used.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/controllers/lifecycle#container-probes

`.spec.pxc.proxysql.readinessProbes.exec`

Description

Exec specifies a command to execute in the container.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.pxc.proxysql.readinessProbes.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

.spec.pxc.proxysql.readinessProbes.exec.command[]

Type

`string`

.spec.pxc.proxysql.readinessProbes.grpc

Description

GRPC specifies a GRPC HealthCheckRequest.

Type

`object`

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

.spec.pxc.proxysql.readinessProbes.httpGet

Description

HTTPGet specifies an HTTP GET request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.

Property	Type	Description
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.pxc.proxysql.readinessProbes.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.pxc.proxysql.readinessProbes.httpGet.httpHeaders[ ]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

name

value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.pxc.proxysql.readinessProbes.tcpSocket

Description

TCPSocket specifies a connection to a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

.spec.pxc.proxysql.resources

Type

object

Property	Type	Description
limits	object	
requests	object	

.spec.pxc.proxysql.resources.limits

Type

object

Property	Type	Description
cpu	string	
ephemeral-storage	string	
memory	string	

.spec.pxc.proxysql.resources.requests

Type

object

Property	Type	Description
cpu	string	
ephemeral-storage	string	
memory	string	

`.spec.pxc.proxysql.serviceAnnotations`

Type

`object`

`.spec.pxc.proxysql.sidecarResources`

Type

`object`

Property	Type	Description
<code>limits</code>	<code>object</code>	
<code>requests</code>	<code>object</code>	

`.spec.pxc.proxysql.sidecarResources.limits`

Type

`object`

Property	Type	Description
<code>cpu</code>	<code>string</code>	
<code>ephemeral-storage</code>	<code>string</code>	
<code>memory</code>	<code>string</code>	

`.spec.pxc.proxysql.sidecarResources.requests`

Type

`object`

Property	Type	Description
<code>cpu</code>	<code>string</code>	
<code>ephemeral-storage</code>	<code>string</code>	
<code>memory</code>	<code>string</code>	

`.spec.pxc.proxysql.startupProbe`

Type

`object`

Property	Type	Description
<code>failureThreshold</code>	<code>integer</code>	
<code>initialDelaySeconds</code>	<code>integer</code>	
<code>periodSeconds</code>	<code>integer</code>	

`.spec.pxc.proxysql.tolerations`

Type

`array`

`.spec.pxc.proxysql.tolerations[]`

Description

The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.

Type

`object`

Property	Type	Description
<code>effect</code>	<code>string</code>	Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.
<code>key</code>	<code>string</code>	Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is empty, operator must be Exists; this combination means to match all values and all keys.
<code>operator</code>	<code>string</code>	Operator represents a key's relationship to the value. Valid operators are Exists and Equal. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category.
<code>tolerationSeconds</code>	<code>integer</code>	TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is not set, which means tolerate the taint forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.
<code>value</code>	<code>string</code>	Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

.spec.pxc.proxysql.volumeSpec

Type

object

Property	Type	Description
<code>emptyDir</code>	object	EmptyDir to use as data volume for mysql. EmptyDir represents a temporary directory that shares a pod's lifetime.
<code>hostPath</code>	object	HostPath to use as data volume for mysql. HostPath represents a pre-existing file or directory on the host machine that is directly exposed to the container.
<code>persistentVolumeClaim</code>	object	PersistentVolumeClaim to specify PVC spec for the volume for mysql data. It has the highest level of precedence, followed by HostPath and EmptyDir. And represents the PVC specification.

.spec.pxc.proxysql.volumeSpec.emptyDir

Description

EmptyDir to use as data volume for mysql. EmptyDir represents a temporary directory that shares a pod's lifetime.

Type

object

Property	Type	Description
<code>medium</code>	<code>string</code>	medium represents what type of storage medium should back this directory. The default is "" which means to use the node's default medium. Must be an empty string (default) or Memory. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir
<code>sizeLimit</code>		sizeLimit is the total amount of local storage required for this EmptyDir volume. The size limit is also applicable for memory medium. The maximum usage on memory medium EmptyDir would be the minimum value between the SizeLimit specified here and the sum of memory limits of all containers in a pod. The default is nil which means that the limit is undefined. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir

.spec.pxc.proxysql.volumeSpec.hostPath

Description

HostPath to use as data volume for mysql. HostPath represents a pre-existing file or directory on the host machine that is directly exposed to the container.

Type

`object`

Required

`path`

Property	Type	Description
path	string	path of the directory on the host. If the path is a symlink, it will follow the link to the real path. More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath ↗
type	string	type for HostPath Volume Defaults to "" More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath ↗

.spec.pxc.proxysql.volumeSpec.persistentVolumeClaim

Description

PersistentVolumeClaim to specify PVC spec for the volume for mysql data. It has the highest level of precedence, followed by HostPath and EmptyDir. And represents the PVC specification.

Type

object

Property	Type	Description
accessModes	array	accessModes contains the desired access modes volume should have. More info: https://kubernetes.io/docs/concepts/storage/persistentvolumes#access-modes-1 ↗
dataSource	object	dataSource field can be used to specify either: An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot)

Property	Type	Description
		An existing PVC (PersistentVolumeClaim) If the provisioner or an external controller can support the specified data source, it will create a new volume based on the contents of the specified data source. When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If the namespace is specified, then dataSourceRef will not be copied to dataSource.
dataSourceRef	object	dataSourceRef specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a PersistentVolumeClaim object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will require the functionality of the dataSource field and as such both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in dataSourceRef, dataSource isn't set to the same value and must be empty. There are three important differences between dataSource and dataSourceRef While dataSource only allows two specific type objects, dataSourceRef allows any non-core object as well as PersistentVolumeClaim objects.

Property	Type	Description
		- While dataSource ignores disallowed values (dropping them), dataSourceRef preserves all values, and generates an error if a disallowed value is specified. - While dataSource only allows local objects, dataSourceRef allows objects in any namespace (Beta) Using this field requires the AnyVolumeDataSource feature gate to be enabled (Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource feature gate to be enabled.
resources	object	resources represents the minimum resources the volume should have. If RecoverVolumeExpansionFailure feature is enabled users are allowed to specify resource requirement that are lower than previous value but must still be higher than capacity recorded in the status field of claim. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources
selector	object	selector is a label query over volumes to consider binding.
storageClassName	string	storageClassName is the name of the StorageClass required by the claim. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#class-1

Property	Type	Description
<code>volumeAttributesClassName</code>	<code>string</code>	<code>volumeAttributesClassName</code> may be used to set the <code>VolumeAttributesClass</code> used by this claim. If specified, the CSI driver will create or update the volume with attributes defined in the corresponding <code>VolumeAttributesClass</code>. This has a different purpose than <code>storageClassName</code>, it can be changed after the claim is created. An empty string value means that <code>VolumeAttributesClass</code> will be applied to the claim if it's not allowed to reset this field to empty string once it is set. If unspecified and the <code>PersistentVolumeClaim</code> is unbound, the default <code>VolumeAttributesClass</code> will be used by the <code>persistentvolume</code> controller if it exists. If the resource referred to by <code>volumeAttributesClass</code> does not exist, this <code>PersistentVolumeClaim</code> will be set to <code>Pending</code> state, as reflected by the <code>modifyVolumeStatus</code> field, until such as a resource exists. More info: https://kubernetes.io/docs/concepts/storage/volume-attributes-classes/ [↗] (Beta) Using this field requires the <code>VolumeAttributesClass</code> feature gate to be enabled (off by default).
<code>volumeMode</code>	<code>string</code>	<code>volumeMode</code> defines what type of volume is required by the claim. Value of <code>Filesystem</code> is implied when not included in claim spec.
<code>volumeName</code>	<code>string</code>	<code>volumeName</code> is the binding reference to the <code>PersistentVolume</code> backing this claim.

`.spec.pxc.proxysql.volumeSpec.persistentVolumeClaim.accessModes`

Description

accessModes contains the desired access modes the volume should have. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1>

Type

array

`.spec.pxc.proxysql.volumeSpec.persistentVolumeClaim.dataSource.accessModes[]`

Type

string

`.spec.pxc.proxysql.volumeSpec.persistentVolumeClaim.dataSource`

Description

dataSource field can be used to specify either: * An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) * An existing PVC (PersistentVolumeClaim) If the provisioner or an external controller can support the specified data source, it will create a new volume based on the contents of the specified data source. When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If the namespace is specified, then dataSourceRef will not be copied to dataSource.

Type

object

Required

kind

name

Property	Type	Description
<code>apiGroup</code>	<code>string</code>	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
<code>kind</code>	<code>string</code>	Kind is the type of resource being referenced
<code>name</code>	<code>string</code>	Name is the name of resource being referenced

`.spec.pxc.proxysql.volumeSpec.persistentVolumeClaim.dataSourceRef`

Description

`dataSourceRef` specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a `PersistentVolumeClaim` object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replace the functionality of the `dataSource` field and as such if both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in `dataSourceRef`, both fields (`dataSource` and `dataSourceRef`) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in `dataSourceRef`, `dataSource` isn't set to the same value and must be empty. There are three important differences between `dataSource` and `dataSourceRef`:

- * While `dataSource` only allows two specific types of objects, `dataSourceRef` allows any non-core object, as well as `PersistentVolumeClaim` objects.
- * While `dataSource` ignores disallowed values (dropping them), `dataSourceRef` preserves all values, and generates an error if a disallowed value is specified.
- * While `dataSource` only allows local objects, `dataSourceRef` allows objects in any namespaces.

(Beta) Using this field requires the `AnyVolumeDataSource` feature gate to be enabled.
 (Alpha) Using the namespace field of `dataSourceRef` requires the `CrossNamespaceVolumeDataSource` feature gate to be enabled.

Type

object

Required

kind name

Property	Type	Description
apiGroup	string	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
kind	string	Kind is the type of resource being referenced
name	string	Name is the name of resource being referenced
namespace	string	Namespace is the namespace of resource being referenced Note that when a namespace is specified, a gateway.networking.k8s.io/ReferenceGrant object is required in the referent namespace to allow that namespace's owner to accept the reference. See the ReferenceGrant documentation for details. (Alpha) This field requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

**.spec.pxc.proxysql.volumeSpec.persistentVolumeClaim.re
sources**

Description

resources represents the minimum resources the volume should have. If RecoverVolumeExpansionFailure feature is enabled users are allowed to specify resource requirements that are lower than previous value but must still be higher than capacity recorded in the status field of the claim. More info:
<https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources>

Type

object

Property	Type	Description
<code>limits</code>	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
<code>requests</code>	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

`.spec.pxc.proxysql.volumeSpec.persistentVolumeClaim.resources.limits`

Description

Limits describes the maximum amount of compute resources allowed. More info:
<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

`.spec.pxc.proxysql.volumeSpec.persistentVolumeClaim.requests`

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info:

<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

`.spec.pxc.proxysql.volumeSpec.persistentVolumeClaim.selector`

Description

selector is a label query over volumes to consider for binding.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.pxc.proxysql.volumeSpec.persistentVolumeClaim.selector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.pxc.proxysql.volumeSpec.persistentVolumeClaim.selector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
<code>key</code>	<code>string</code>	key is the label key that the selector applies to.
<code>operator</code>	<code>string</code>	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.

Property	Type	Description
<code>values</code>	<code>array</code>	<code>values</code> is an array of string values. If the operator is <code>In</code> or <code>NotIn</code> , the <code>values</code> array must be non-empty. If the operator is <code>Exists</code> or <code>DoesNotExist</code> , the <code>values</code> array must be empty. This array is replaced during a strategic merge patch.

`.spec.pxc.proxysql.volumeSpec.persistentVolumeClaim.selector.matchExpressions[].values`

Description

`values` is an array of string values. If the operator is `In` or `NotIn`, the `values` array must be non-empty. If the operator is `Exists` or `DoesNotExist`, the `values` array must be empty. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.pxc.proxysql.volumeSpec.persistentVolumeClaim.selector.matchExpressions[].values[]`

Type

`string`

`.spec.pxc.proxysql.volumeSpec.persistentVolumeClaim.selector.matchLabels`

Description

`matchLabels` is a map of {key,value} pairs. A single {key,value} in the `matchLabels` map is equivalent to an element of `matchExpressions`, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.pxc.pxc

Type

object

Property	Type	Description
affinity	object	
annotations	object	
autoRecovery	boolean	
configuration	string	
containerSecurityContext	object	SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.
enableProbes	boolean	
enabled	boolean	
externalTrafficPolicy	string	for backwards compat
forceUnsafeBootstrap	boolean	
gracePeriod	integer	

Property	Type	Description
<code>image</code>	<code>string</code>	
<code>imagePullPolicy</code>	<code>string</code>	PullPolicy describes a policy for if/when to pull a container image
<code>imagePullSecrets</code>	<code>array</code>	
<code>labels</code>	<code>object</code>	
<code>livenessDelaySec</code>	<code>integer</code>	
<code>livenessProbes</code>	<code>object</code>	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
<code>loadBalancerSourceRanges</code>	<code>array</code>	
<code>nodeSelector</code>	<code>object</code>	
<code>podDisruptionBudget</code>	<code>object</code>	
<code>podSecurityContext</code>	<code>object</code>	PodSecurityContext holds pod-level security attributes and common container settings. Some fields are also present in container.securityContext. Field values of container.securityContext take precedence over field values of PodSecurityContext.
<code>priorityClassName</code>	<code>string</code>	

Property	Type	Description
<code>readinessDelaySec</code>	<code>integer</code>	
<code>readinessProbes</code>	<code>object</code>	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
<code>replicasExternalTrafficPolicy</code>	<code>string</code>	for backwards compat
<code>replicasServiceType</code>	<code>string</code>	Service Type string describes ingress methods for a service
<code>replicationChannels</code>	<code>array</code>	
<code>resources</code>	<code>object</code>	
<code>schedulerName</code>	<code>string</code>	
<code>serverMaxConnections</code>	<code>integer</code>	
<code>serviceAccountName</code>	<code>string</code>	
<code>serviceAnnotations</code>	<code>object</code>	
<code>serviceType</code>	<code>string</code>	Service Type string describes ingress methods for a service
<code>sidecarResources</code>	<code>object</code>	
<code>size</code>	<code>integer</code>	

Property	Type	Description
<code>sslInternalSecretName</code>	<code>string</code>	
<code>sslSecretName</code>	<code>string</code>	
<code>startupProbe</code>	<code>object</code>	
<code>tolerations</code>	<code>array</code>	
<code>vaultSecretName</code>	<code>string</code>	
<code>volumeSpec</code>	<code>object</code>	
<code>writeNode</code>	<code>string</code>	

`.spec.pxc.pxc.affinity`

Type

`object`

Property	Type	Description
<code>advanced</code>	<code>object</code>	Affinity is a group of affinity scheduling rules.
<code>antiAffinityTopologyKey</code>	<code>string</code>	

`.spec.pxc.pxc.affinity.advanced`

Description

Affinity is a group of affinity scheduling rules.

Type

`object`

Property	Type	Description
<code>nodeAffinity</code>	<code>object</code>	Describes node affinity scheduling rules for the pod.
<code>podAffinity</code>	<code>object</code>	Describes pod affinity scheduling rules (e.g. co-locate this pod in the same node, zone, etc. as some other pod(s)).
<code>podAntiAffinity</code>	<code>object</code>	Describes pod anti-affinity scheduling rules (e.g. avoid putting this pod in the same node, zone, etc. as some other pod(s)).

`.spec.pxc.pxc.affinity.advanced.nodeAffinity`

Description

Describes node affinity scheduling rules for the pod.

Type

`object`

Property	Type	Description
<code>preferredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most

Property	Type	Description
		preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution affinity expressions, etc.) compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>object</code>	If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to an update), the system may or may not try to eventually evict the pod from its node.

`.spec.pxc.pxc.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest sum are the most preferred.

Type

array

`.spec.pxc.pxc.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

An empty preferred scheduling term matches all objects with implicit weight 0 (i.e. it's a no-op). A null preferred scheduling term matches no objects (i.e. is also a no-op).

Type

object

Required

preference

weight

Property	Type	Description
preference	object	A node selector term, associated with the corresponding weight.

Property	Type	Description
<code>weight</code>	<code>integer</code>	Weight associated with matching the corresponding nodeSelectorTerm, in the range 1-100.

`.spec.pxc.pxc.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference`

Description

A node selector term, associated with the corresponding weight.

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	A list of node selector requirements by node's labels.
<code>matchFields</code>	<code>array</code>	A list of node selector requirements by node's fields.

`.spec.pxc.pxc.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions`

Description

A list of node selector requirements by node's labels.

Type

`array`

.spec.pxc.pxc.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[]

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.pxc.pxc.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.match

Expressions[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.pxc.pxc.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values[]

Type

string

.spec.pxc.pxc.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields

Description

A list of node selector requirements by node's fields.

Type

array

.spec.pxc.pxc.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[]

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.pxc.pxc.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.pxc.pxc.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values[]

Type

string

.spec.pxc.pxc.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to an update), the system may or may not try to eventually evict the pod from its node.

Type

object

Required

nodeSelectorTerms

Property	Type	Description
<code>nodeSelectorTerms</code>	array	Required. A list of node selector terms. The terms are ORed.

.spec.pxc.pxc.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerm

S

Description

Required. A list of node selector terms. The terms are ORed.

Type

array

.spec.pxc.pxc.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[]

Description

A null or empty node selector term matches no objects. The requirements of them are ANDed. The TopologySelectorTerm type implements a subset of the NodeSelectorTerm.

Type

object

Property	Type	Description
matchExpressions	array	A list of node selector requirements by node's labels.
matchFields	array	A list of node selector requirements by node's fields.

.spec.pxc.pxc.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions

Description

A list of node selector requirements by node's labels.

Type

array

.spec.pxc.pxc.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[]

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.pxc.pxc.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.pxc.pxc.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values[]

Type

string

.spec.pxc.pxc.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields

Description

A list of node selector requirements by node's fields.

Type

array

.spec.pxc.pxc.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerm

s[].matchFields[]

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.pxc.pxc.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.pxc.pxc.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values[]

Type

string

.spec.pxc.pxc.affinity.advanced.podAffinity

Description

Describes pod affinity scheduling rules (e.g. co-locate this pod in the same node, zone, etc. as some other pod(s)).

Type

object

Property	Type	Description
<code>preferredDuringSchedulingIgnoredDuringExecution</code>	array	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most

Property	Type	Description
		preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	If the affinity requirement: specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirement: specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When

Property	Type	Description
		there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

.spec.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

array

.spec.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

Description

The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Type

object

Required

`podAffinityTerm``weight`

Property	Type	Description
<code>podAffinityTerm</code>	<code>object</code>	Required. A pod affinity term, associated with the corresponding weight.
<code>weight</code>	<code>integer</code>	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

`.spec.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm`

Description

Required. A pod affinity term, associated with the corresponding weight.

Type

`object`

Required

`topologyKey`

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys

Property	Type	Description
		are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both <code>matchLabelKeys</code> and <code>labelSelector</code>. Also, <code>matchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. This is a beta field and requires enabling <code>MatchLabelKeysInPodAffinity</code> feature gate (enabled by default).
<code>mismatchLabelKeys</code>	<code>array</code>	<code>MismatchLabelKeys</code> is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both <code>mismatchLabelKeys</code> and <code>labelSelector</code>. Also, <code>mismatchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. This is a beta field and requires enabling <code>MatchLabelKeysInPodAffinity</code> feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the <code>namespaces</code> field. null selector and null or

Property	Type	Description
		empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
namespaces	array	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
topologyKey	string	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

.spec.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

object

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

labelSelector.matchExpressions[].values[]

Type

string

.spec.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]

Type

string

.spec.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]

Type

string

.spec.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
matchExpressions	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
matchLabels	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces.matchExpressions

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces.selector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces.selector.matchExpressions[]`

mespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[0].mespaceSelector.matchExpressions[].values[]

Type

string

.spec.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[0].mespaceSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

string

.spec.pxc.pxc.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Type

array

.spec.pxc.pxc.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

Defines a set of pods (namely those matching the `labelSelector` relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key `<topologyKey>` matches that of any node on which a pod of the set of pods is running

Type

object

Required

topologyKey

Property	Type	Description
<code>labelSelector</code>	object	A label query over a set of resources, in this case pods. If it's null, this <code>PodAffinityTerm</code> matches with no Pods.
<code>matchLabelKeys</code>	array	<code>MatchLabelKeys</code> is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both <code>matchLabelKeys</code> and <code>labelSelector</code> . Also, <code>matchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. This is a beta field and requires enabling <code>MatchLabelKeysInPodAffinity</code> feature gate (enabled by default).
<code>mismatchLabelKeys</code>	array	<code>MismatchLabelKeys</code> is a set of pod label keys to select which pods will be taken into consideration. The

Property	Type	Description
		keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both <code>mismatchLabelKeys</code> and <code>labelSelector</code>. Also, <code>mismatchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. This is a beta field and requires enabling <code>MatchLabelKeysInPodAffinity</code> feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the <code>namespaces</code> field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector (<code>{}</code>) matches all namespaces.
<code>namespaces</code>	<code>array</code>	<code>namespaces</code> specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by <code>namespaceSelector</code>. null or empty namespaces list and null <code>namespaceSelector</code> means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the <code>labelSelector</code> in the specified namespaces, where co-

Property	Type	Description
		located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.pxc.pxc.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.pxc.pxc.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.pxc.pxc.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.pxc.pxc.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.pxc.pxc.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]`

Type

string

`.spec.pxc.pxc.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

`.spec.pxc.pxc.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys`

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

`.spec.pxc.pxc.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]`

Type

string

`.spec.pxc.pxc.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys`

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector.

Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.pxc.pxc.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]

Type

string

.spec.pxc.pxc.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an

Property	Type	Description
		element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.pxc.pxc.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.pxc.pxc.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.pxc.pxc.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.pxc.pxc.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]

Type

`string`

.spec.pxc.pxc.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

`object`

.spec.pxc.pxc.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

`array`

.spec.pxc.pxc.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]

Type

`string`

.spec.pxc.pxc.affinity.advanced.podAntiAffinity

Description

Describes pod anti-affinity scheduling rules (e.g. avoid putting this pod in the same node, zone, etc. as some other pod(s)).

Type

object

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution, anti-affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest

Property	Type	Description
		sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights,

i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling anti-affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

array

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

Description

The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Type

object

Required

podAffinityTerm weight

Property	Type	Description
podAffinityTerm	object	Required. A pod affinity term, associated with the corresponding weight.
weight	integer	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTer

m

Description

Required. A pod affinity term, associated with the corresponding weight.

Type

object

Required

topologyKey

Property	Type	Description
<code>labelSelector</code>	object	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	array	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>mismatchLabelKeys</code>	array	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The

Property	Type	Description
		keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both <code>mismatchLabelKeys</code> and <code>labelSelector</code>. Also, <code>mismatchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. This is a beta field and requires enabling <code>MatchLabelKeysInPodAffinity</code> feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the <code>namespaces</code> field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector (<code>{}</code>) matches all namespaces.
<code>namespaces</code>	<code>array</code>	<code>namespaces</code> specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by <code>namespaceSelector</code>. null or empty namespaces list and null <code>namespaceSelector</code> means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the <code>labelSelector</code> in the specified namespaces, where co-

Property	Type	Description
		located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm`

m.labelSelector.matchExpressions

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.m.labelSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.

Property	Type	Description
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]

Type

string

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]

Type

string

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm

m.mismatchLabelKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]

Type

string

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm

m.namespaceSelector.matchExpressions[].values[]

Type

string

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

string

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Type

array

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running

Type

object

Required

topologyKey

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field

Property	Type	Description
		and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

`array`

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]

Type

`string`

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

`object`

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also,

matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]

Type

string

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]

Type

string

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]

Type

string

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]

Type

string

.spec.pxc.pxc.annotations

Type

object

.spec.pxc.pxc.containerSecurityContext

Description

SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.

Type

object

Property	Type	Description
<code>allowPrivilegeEscalation</code>	<code>boolean</code>	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the <code>no_new_privs</code> flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: 1. run as Privileged 2. has CAP_SYS_ADMIN Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>appArmorProfile</code>	<code>object</code>	<code>appArmorProfile</code> is the AppArmor options to use by this container. If set, this profile overrides the pod's <code>appArmorProfile</code>. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>capabilities</code>	<code>object</code>	The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>privileged</code>	<code>boolean</code>	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when <code>spec.os.name</code> is windows.

Property	Type	Description
<code>procMount</code>	<code>string</code>	<code>procMount</code> denotes the type of proc mount to use for the containers. The default value is <code>Default</code> which uses the container runtime defaults for readonly paths and masked paths. This requires the <code>ProcMountType</code> feature flag to be enabled. Note that this field cannot be set when <code>spec.os.name</code> is <code>windows</code>.
<code>readOnlyRootFilesystem</code>	<code>boolean</code>	Whether this container has a read-only root filesystem. Default is <code>false</code>. Note that this field cannot be set when <code>spec.os.name</code> is <code>windows</code>.
<code>runAsGroup</code>	<code>integer</code>	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in <code>PodSecurityContext</code>. If set in both <code>SecurityContext</code> and <code>PodSecurityContext</code>, the value specified in <code>SecurityContext</code> takes precedence. Note that this field cannot be set when <code>spec.os.name</code> is <code>windows</code>.
<code>runAsNonRoot</code>	<code>boolean</code>	Indicates that the container must run as a non-root user. If <code>true</code>, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or <code>false</code>, no such validation will be performed. May also be set in <code>PodSecurityContext</code>. If set in

Property	Type	Description
		both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seccompProfile	object	The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
<code>windowsOptions</code>	<code>object</code>	The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

.spec.pxc.pxc.containerSecurityContext.appArmorProfile

Description

appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
<code>type</code>	<code>string</code>	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-

Property	Type	Description
		loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

.spec.pxc.pxc.containerSecurityContext.capabilities

Description

The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when spec.os.name is windows.

Type

object

Property	Type	Description
add	array	Added capabilities
drop	array	Removed capabilities

.spec.pxc.pxc.containerSecurityContext.capabilities.add

Description

Added capabilities

Type

array

.spec.pxc.pxc.containerSecurityContext.capabilities.add[]

Description

Capability represent POSIX capabilities type

Type

string

.spec.pxc.pxc.containerSecurityContext.capabilities.drop

Description

Removed capabilities

Type

array

.spec.pxc.pxc.containerSecurityContext.capabilities.drop[]

Description

Capability represent POSIX capabilities type

Type

string

.spec.pxc.pxc.containerSecurityContext.seLinuxOptions

Description

The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Type

object

Property	Type	Description
<code>level</code>	<code>string</code>	Level is SELinux level label that applies to the container.
<code>role</code>	<code>string</code>	Role is a SELinux role label that applies to the container.
<code>type</code>	<code>string</code>	Type is a SELinux type label that applies to the container.
<code>user</code>	<code>string</code>	User is a SELinux user label that applies to the container.

`.spec.pxc.pxc.containerSecurityContext.seccompProfile`

Description

The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when `spec.os.name` is windows.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	localhostProfile indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured

Property	Type	Description
		seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.
		type indicates which kind of seccomp profile will be applied. Valid options are:
type	string	Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied.

.spec.pxc.pxc.containerSecurityContext.windowsOptions

Description

The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
gmsaCredentialSpec	string	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the GMSACredentialSpecName field.

Property	Type	Description
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

`.spec.pxc.pxc.imagePullSecrets`

Type

`array`

`.spec.pxc.pxc.imagePullSecrets[]`

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

.spec.pxc.pxc.labels

Type

object

.spec.pxc.pxc.livenessProbes

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	Exec specifies a command to execute in the

Property	Type	Description
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe to be considered failed after having succeeded. Default value is 3. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies a GRPC HealthCheckRequest
<code>httpGet</code>	<code>object</code>	HTTPGet specifies an HTTP GET request to
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe. Default is 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe to be considered successful after having failed. Default is 1. Must be 1 for liveness and startup. Minimum
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies a connection to a TCP
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod needs to gracefully shutdown upon probe failure. The grace period is the duration in seconds after the processes running

Property	Type	Description
		pod are sent a termination signal and the timer processes are forcibly halted with a kill signal. If the value is longer than the expected cleanup time process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided in the pod spec. Value must be non-negative integer. value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. spec.terminationGracePeriodSeconds is used.
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/controllers/lifecycle#container-probes

.spec.pxc.pxc.livenessProbes.exec

Description

Exec specifies a command to execute in the container.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a

Property	Type	Description
		shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.pxc.pxc.livenessProbes.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

.spec.pxc.pxc.livenessProbes.exec.command[]

Type

string

.spec.pxc.pxc.livenessProbes.grpc

Description

GRPC specifies a GRPC HealthCheckRequest.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

.spec.pxc.pxc.livenessProbes.httpGet

Description

HTTPGet specifies an HTTP GET request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.

Property	Type	Description
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>	<code>int</code>	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.pxc.pxc.livenessProbes.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.pxc.pxc.livenessProbes.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name`

`value`

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.pxc.pxc.livenessProbes.tcpSocket`

Description

TCPSocket specifies a connection to a TCP port.

Type

`object`

Required

`port`

Property	Type	Description
<code>host</code>	<code>string</code>	Optional: Host name to connect to, defaults to the pod IP.
<code>port</code>		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.pxc.pxc.loadBalancerSourceRanges`

Type

`array`

.spec.pxc.pxc.loadBalancerSourceRanges[]

Type

string

.spec.pxc.pxc.nodeSelector

Type

object

.spec.pxc.pxc.podDisruptionBudget

Type

object

Property	Type	Description
maxUnavailable		
minAvailable		

.spec.pxc.pxc.podSecurityContext

Description

PodSecurityContext holds pod-level security attributes and common container settings. Some fields are also present in container.securityContext. Field values of container.securityContext take precedence over field values of PodSecurityContext.

Type

object

Property	Type	Description
appArmorProfile	object	appArmorProfile is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.
fsGroup	integer	A special supplemental group that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod: 1. The owning GID will be the FSGroup 2. The setgid bit is set (new files created in the volume will be owned by FSGroup) 3. The permission bits are OR'd with rw-rw-- -- If unset, the Kubelet will not modify the ownership and permissions of any volume. Note that this field cannot be set when spec.os.name is windows.
fsGroupChangePolicy	string	fsGroupChangePolicy defines behavior of changing ownership and permission of the volume before being exposed inside Pod. This field will only apply to volume types which support fsGroup based ownership(and permissions). It will have no effect on ephemeral volume types such as: secret, configmaps and emptydir. Valid values are "OnRootMismatch" and "Always". If not specified, "Always" is used. Note that this

Property	Type	Description
		field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this

Property	Type	Description
		field cannot be set when spec.os.name is windows.
seLinuxChangePolicy	string	seLinuxChangePolicy defines how the container's SELinux label is applied to all volumes used by the Pod. It has no effect on nodes that do not support SELinux or to volumes does not support SELinux. Valid values are "MountOption" and "Recursive". "Recursive" means relabeling of all files on all Pod volumes by the container runtime. This may be slow for large volumes, but allows mixing privileged and unprivileged Pods sharing the same volume on the same node. "MountOption" mounts all eligible Pod volumes with <code>-o context</code> mount option. This requires all Pods that share the same volume to use the same SELinux label. It is not possible to share the same volume among privileged and unprivileged Pods. Eligible volumes are in-tree FibreChannel and iSCSI volumes, and all CSI volumes whose CSI driver announces SELinux support by setting spec.seLinuxMount: true in their CSIDriver instance. Other volumes are always re-labelled recursively. "MountOption" value is allowed only when SELinuxMount feature gate is enabled. If not specified and SELinuxMount feature gate is enabled, "MountOption" is used. If not specified and SELinuxMount feature

Property	Type	Description
		gate is disabled, "MountOption" is used for ReadWriteOncePod volumes and "Recursive" for all other volumes. This field affects only Pods that have SELinux label set, either in PodSecurityContext or in SecurityContext of all containers. All Pods that use the same volume should use the same seLinuxChangePolicy, otherwise some pods can get stuck in ContainerCreating state. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
seccompProfile	object	The seccomp options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
<code>supplementalGroups</code>	<code>array</code>	A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.
<code>supplementalGroupsPolicy</code>	<code>string</code>	Defines how supplemental groups of the first container processes are calculated. Valid values are "Merge" and "Strict". If not specified, "Merge" is used. (Alpha) Using the field requires the SupplementalGroupsPolicy feature gate to be enabled and the container runtime must implement support for this feature. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
<code>sysctls</code>	<code>array</code>	Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.
<code>windowsOptions</code>	<code>object</code>	The Windows specific settings applied to all containers. If unspecified, the options within a container's SecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

`.spec.pxc.pxc.podSecurityContext.appArmorProfile`

Description

appArmorProfile is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured

Property	Type	Description
		on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
<code>type</code>	<code>string</code>	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

.spec.pxc.pxc.podSecurityContext.seLinuxOptions

Description

The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Property	Type	Description
<code>level</code>	<code>string</code>	Level is SELinux level label that applies to the container.
<code>role</code>	<code>string</code>	Role is a SELinux role label that applies to the container.
<code>type</code>	<code>string</code>	Type is a SELinux type label that applies to the container.

Property	Type	Description
<code>user</code>	<code>string</code>	User is a SELinux user label that applies to the container.

`.spec.pxc.pxc.podSecurityContext.seccompProfile`

Description

The seccomp options to use by the containers in this pod. Note that this field cannot be set when `spec.os.name` is `windows`.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.
<code>type</code>	<code>string</code>	<code>type</code> indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied.

.spec.pxc.pxc.podSecurityContext.supplementalGroups

Description

A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.

Type

array

.spec.pxc.pxc.podSecurityContext.supplementalGroups[]

Type

integer

.spec.pxc.pxc.podSecurityContext.sysctls

Description

Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.

Type

array

.spec.pxc.pxc.podSecurityContext.sysctls[]

Description

Sysctl defines a kernel parameter to be set

Type

object

Required

name

value

Property	Type	Description
name	string	Name of a property to set
value	string	Value of a property to set

.spec.pxc.pxc.podSecurityContext.windowsOptions

Description

The Windows specific settings applied to all containers. If unspecified, the options within a container's SecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
gmsaCredentialSpec	string	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.

Property	Type	Description
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

.spec.pxc.pxc.readinessProbes

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

`object`

Property	Type	Description
<code>exec</code>	<code>object</code>	Exec specifies a command to execute in the
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe to be considered failed after having succeeded. Default value is 3. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies a GRPC HealthCheckRequest to the service.
<code>httpGet</code>	<code>object</code>	HTTPGet specifies an HTTP GET request to the service.
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/controllers/lifecycle#container-probes
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe to be considered successful after having failed. Default is 1. Must be 1 for liveness and startup. Minimum value is 1.
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies a connection to a TCP endpoint.

Property	Type	Description
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod needs gracefully upon probe failure. The grace period duration in seconds after the processes running in the pod are sent a termination signal and the timer begins. Once the grace period has elapsed, the processes are forcibly halted with a kill signal. A value greater than zero indicates a grace period value longer than the expected cleanup time for the process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided in the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. <code>spec.terminationGracePeriodSeconds</code> is used.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/controllers/lifecycle#container-probes

`.spec.pxc.pxc.readinessProbes.exec`

Description

Exec specifies a command to execute in the container.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.pxc.pxc.readinessProbes.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

`.spec.pxc.pxc.readinessProbes.exec.command[]`

Type

`string`

`.spec.pxc.pxc.readinessProbes.grpc`

Description

GRPC specifies a GRPC HealthCheckRequest.

Type

`object`

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

.spec.pxc.pxc.readinessProbes.httpGet

Description

HTTPGet specifies an HTTP GET request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.

Property	Type	Description
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.pxc.pxc.readinessProbes.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.pxc.pxc.readinessProbes.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name`

`value`

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.pxc.pxc.readinessProbes.tcpSocket`

Description

TCPSocket specifies a connection to a TCP port.

Type

`object`

Required

`port`

Property	Type	Description
<code>host</code>	<code>string</code>	Optional: Host name to connect to, defaults to the pod IP.
<code>port</code>		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.pxc.pxc.replicationChannels`

Type

`array`

`.spec.pxc.pxc.replicationChannels[]`

Type

`object`

Property	Type	Description
<code>configuration</code>	<code>object</code>	
<code>enabled</code>	<code>boolean</code>	
<code>name</code>	<code>string</code>	
<code>sourceProxyList</code>	<code>array</code>	
<code>sourcesList</code>	<code>array</code>	

`.spec.pxc.pxc.replicationChannels[].configuration`

Type

`object`

Property	Type	Description
<code>sourceConnectRetry</code>	<code>integer</code>	
<code>sourceRetryCount</code>	<code>integer</code>	

`.spec.pxc.pxc.replicationChannels[].sourceProxyList`

Type

`array`

`.spec.pxc.pxc.replicationChannels[].sourceProxyList[]`

Type

object

Required

host hostName

Property	Type	Description
host	string	
hostName	string	
port	integer	
weight	integer	

.spec.pxc.pxc.replicationChannels[].sourcesList

Type

array

.spec.pxc.pxc.replicationChannels[].sourcesList[]

Type

object

Required

host hostName

Property	Type	Description
host	string	
hostName	string	
port	integer	
weight	integer	

.spec.pxc.pxc.resources

Type

object

Property	Type	Description
limits	object	
requests	object	

.spec.pxc.pxc.resources.limits

Type

object

Property	Type	Description
cpu	string	
ephemeral-storage	string	
memory	string	

.spec.pxc.pxc.resources.requests

Type

object

Property	Type	Description
cpu	string	
ephemeral-storage	string	
memory	string	

`.spec.pxc.pxc.serviceAnnotations`

Type

`object`

`.spec.pxc.pxc.sidecarResources`

Type

`object`

Property	Type	Description
<code>limits</code>	<code>object</code>	
<code>requests</code>	<code>object</code>	

`.spec.pxc.pxc.sidecarResources.limits`

Type

`object`

Property	Type	Description
<code>cpu</code>	<code>string</code>	
<code>ephemeral-storage</code>	<code>string</code>	
<code>memory</code>	<code>string</code>	

`.spec.pxc.pxc.sidecarResources.requests`

Type

`object`

Property	Type	Description
<code>cpu</code>	<code>string</code>	
<code>ephemeral-storage</code>	<code>string</code>	
<code>memory</code>	<code>string</code>	

`.spec.pxc.pxc.startupProbe`

Type

`object`

Property	Type	Description
<code>failureThreshold</code>	<code>integer</code>	
<code>initialDelaySeconds</code>	<code>integer</code>	
<code>periodSeconds</code>	<code>integer</code>	

`.spec.pxc.pxc.tolerations`

Type

`array`

`.spec.pxc.pxc.tolerations[]`

Description

The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.

Type

`object`

Property	Type	Description
<code>effect</code>	<code>string</code>	Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.
<code>key</code>	<code>string</code>	Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is empty, operator must be Exists; this combination means to match all values and all keys.
<code>operator</code>	<code>string</code>	Operator represents a key's relationship to the value. Valid operators are Exists and Equal. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category.
<code>tolerationSeconds</code>	<code>integer</code>	TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is not set, which means tolerate the taint forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.
<code>value</code>	<code>string</code>	Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

.spec.pxc.pxc.volumeSpec

Type

object

Property	Type	Description
<code>emptyDir</code>	object	EmptyDir to use as data volume for mysql. EmptyDir represents a temporary directory that shares a pod's lifetime.
<code>hostPath</code>	object	HostPath to use as data volume for mysql. HostPath represents a pre-existing file or directory on the host machine that is directly exposed to the container.
<code>persistentVolumeClaim</code>	object	PersistentVolumeClaim to specify PVC spec for the volume for mysql data. It has the highest level of precedence, followed by HostPath and EmptyDir. And represents the PVC specification.

.spec.pxc.pxc.volumeSpec.emptyDir

Description

EmptyDir to use as data volume for mysql. EmptyDir represents a temporary directory that shares a pod's lifetime.

Type

object

Property	Type	Description
<code>medium</code>	<code>string</code>	medium represents what type of storage medium should back this directory. The default is "" which means to use the node's default medium. Must be an empty string (default) or Memory. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir
<code>sizeLimit</code>		sizeLimit is the total amount of local storage required for this EmptyDir volume. The size limit is also applicable for memory medium. The maximum usage on memory medium EmptyDir would be the minimum value between the SizeLimit specified here and the sum of memory limits of all containers in a pod. The default is nil which means that the limit is undefined. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir

`.spec.pxc.pxc.volumeSpec.hostPath`

Description

HostPath to use as data volume for mysql. HostPath represents a pre-existing file or directory on the host machine that is directly exposed to the container.

Type

`object`

Required

`path`

Property	Type	Description
path	string	path of the directory on the host. If the path is a symlink, it will follow the link to the real path. More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath ↗
type	string	type for HostPath Volume Defaults to "" More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath ↗

.spec.pxc.pxc.volumeSpec.persistentVolumeClaim

Description

PersistentVolumeClaim to specify PVC spec for the volume for mysql data. It has the highest level of precedence, followed by HostPath and EmptyDir. And represents the PVC specification.

Type

object

Property	Type	Description
accessModes	array	accessModes contains the desired access modes volume should have. More info: https://kubernetes.io/docs/concepts/storage/persistentvolumes#access-modes-1 ↗
dataSource	object	dataSource field can be used to specify either: An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot)

Property	Type	Description
		An existing PVC (PersistentVolumeClaim) If the provisioner or an external controller can support the specified data source, it will create a new volume based on the contents of the specified data source. When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If the namespace is specified, then dataSourceRef will not be copied to dataSource.
dataSourceRef	object	dataSourceRef specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a PersistentVolumeClaim object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will require the functionality of the dataSource field and as such both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in dataSourceRef, dataSource isn't set to the same value and must be empty. There are three important differences between dataSource and dataSourceRef While dataSource only allows two specific type objects, dataSourceRef allows any non-core object as well as PersistentVolumeClaim objects.

Property	Type	Description
		- While dataSource ignores disallowed values (dropping them), dataSourceRef preserves all values, and generates an error if a disallowed value is specified. - While dataSource only allows local objects, dataSourceRef allows objects in any namespace (Beta) Using this field requires the AnyVolumeDataSource feature gate to be enabled (Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource feature gate to be enabled.
resources	object	resources represents the minimum resources the volume should have. If RecoverVolumeExpansionFailure feature is enabled users are allowed to specify resource requirement that are lower than previous value but must still be higher than capacity recorded in the status field of claim. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources
selector	object	selector is a label query over volumes to consider binding.
storageClassName	string	storageClassName is the name of the StorageClass required by the claim. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#class-1

Property	Type	Description
<code>volumeAttributesClassName</code>	<code>string</code>	<code>volumeAttributesClassName</code> may be used to set the <code>VolumeAttributesClass</code> used by this claim. If specified, the CSI driver will create or update the volume with attributes defined in the corresponding <code>VolumeAttributesClass</code>. This has a different purpose than <code>storageClassName</code>, it can be changed after the claim is created. An empty string value means that <code>VolumeAttributesClass</code> will be applied to the claim if it's not allowed to reset this field to empty string once it is set. If unspecified and the <code>PersistentVolumeClaim</code> is unbound, the default <code>VolumeAttributesClass</code> will be used by the <code>persistentvolume</code> controller if it exists. If the resource referred to by <code>volumeAttributesClass</code> does not exist, this <code>PersistentVolumeClaim</code> will be set to <code>Pending</code> state, as reflected by the <code>modifyVolumeStatus</code> field, until such as a resource exists. More info: https://kubernetes.io/docs/concepts/storage/volume-attributes-classes/ (Beta) Using this field requires the <code>VolumeAttributesClass</code> feature gate to be enabled (off by default).
<code>volumeMode</code>	<code>string</code>	<code>volumeMode</code> defines what type of volume is required by the claim. Value of <code>Filesystem</code> is implied when not included in claim spec.
<code>volumeName</code>	<code>string</code>	<code>volumeName</code> is the binding reference to the <code>PersistentVolume</code> backing this claim.

.spec.pxc.pxc.volumeSpec.persistentVolumeClaim.accessModes

Description

accessModes contains the desired access modes the volume should have. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1>

Type

array

.spec.pxc.pxc.volumeSpec.persistentVolumeClaim.accessModes[]

Type

string

.spec.pxc.pxc.volumeSpec.persistentVolumeClaim.dataSource

Description

dataSource field can be used to specify either: * An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) * An existing PVC (PersistentVolumeClaim) If the provisioner or an external controller can support the specified data source, it will create a new volume based on the contents of the specified data source. When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If the namespace is specified, then dataSourceRef will not be copied to dataSource.

Type

object

Required

kind

name

Property	Type	Description
<code>apiGroup</code>	<code>string</code>	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
<code>kind</code>	<code>string</code>	Kind is the type of resource being referenced
<code>name</code>	<code>string</code>	Name is the name of resource being referenced

`.spec.pxc.pxc.volumeSpec.persistentVolumeClaim.dataSourceRef`

Description

`dataSourceRef` specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a `PersistentVolumeClaim` object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replace the functionality of the `dataSource` field and as such if both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in `dataSourceRef`, both fields (`dataSource` and `dataSourceRef`) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in `dataSourceRef`, `dataSource` isn't set to the same value and must be empty. There are three important differences between `dataSource` and `dataSourceRef`:

- * While `dataSource` only allows two specific types of objects, `dataSourceRef` allows any non-core object, as well as `PersistentVolumeClaim` objects.
- * While `dataSource` ignores disallowed values (dropping them), `dataSourceRef` preserves all values, and generates an error if a disallowed value is specified.
- * While `dataSource` only allows local objects, `dataSourceRef` allows objects in any namespaces.

(Beta) Using this field requires the `AnyVolumeDataSource` feature gate to be enabled.
 (Alpha) Using the namespace field of `dataSourceRef` requires the `CrossNamespaceVolumeDataSource` feature gate to be enabled.

Type

object

Required

kind name

Property	Type	Description
apiGroup	string	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
kind	string	Kind is the type of resource being referenced
name	string	Name is the name of resource being referenced
namespace	string	Namespace is the namespace of resource being referenced Note that when a namespace is specified, a gateway.networking.k8s.io/ReferenceGrant object is required in the referent namespace to allow that namespace's owner to accept the reference. See the ReferenceGrant documentation for details. (Alpha) This field requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

.spec.pxc.pxc.volumeSpec.persistentVolumeClaim.resources

Description

resources represents the minimum resources the volume should have. If RecoverVolumeExpansionFailure feature is enabled users are allowed to specify resource requirements that are lower than previous value but must still be higher than capacity recorded in the status field of the claim. More info:
<https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources>

Type

object

Property	Type	Description
<code>limits</code>	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
<code>requests</code>	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

`.spec.pxc.pxc.volumeSpec.persistentVolumeClaim.resources.limits`

Description

Limits describes the maximum amount of compute resources allowed. More info:
<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

`.spec.pxc.pxc.volumeSpec.persistentVolumeClaim.resources.requests`

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info:

<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

`.spec.pxc.pxc.volumeSpec.persistentVolumeClaim.selector`

Description

selector is a label query over volumes to consider for binding.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.pxc.pxc.volumeSpec.persistentVolumeClaim.selector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.pxc.pxc.volumeSpec.persistentVolumeClaim.selector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
<code>key</code>	<code>string</code>	key is the label key that the selector applies to.
<code>operator</code>	<code>string</code>	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.

Property	Type	Description
<code>values</code>	<code>array</code>	<code>values</code> is an array of string values. If the operator is <code>In</code> or <code>NotIn</code> , the values array must be non-empty. If the operator is <code>Exists</code> or <code>DoesNotExist</code> , the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.pxc.pxc.volumeSpec.persistentVolumeClaim.selector.matchExpressions[].values`

Description

`values` is an array of string values. If the operator is `In` or `NotIn`, the values array must be non-empty. If the operator is `Exists` or `DoesNotExist`, the values array must be empty. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.pxc.pxc.volumeSpec.persistentVolumeClaim.selector.matchExpressions[].values[]`

Type

`string`

`.spec.pxc.pxc.volumeSpec.persistentVolumeClaim.selector.matchLabels`

Description

`matchLabels` is a map of {key,value} pairs. A single {key,value} in the `matchLabels` map is equivalent to an element of `matchExpressions`, whose key field is "key", the operator is `In`, and the values array contains only "value". The requirements are ANDed.

Type

`object`

`.spec.pxc.serviceMonitor`

Type

`object`

Property	Type	Description
<code>customMetricRelabelings</code>	<code>boolean</code>	
<code>enabled</code>	<code>boolean</code>	
<code>interval</code>	<code>string</code>	
<code>labels</code>	<code>object</code>	
<code>metricRelabelings</code>	<code>array</code>	
<code>scrapeTimeout</code>	<code>string</code>	

`.spec.pxc.serviceMonitor.labels`

Type

`object`

`.spec.pxc.serviceMonitor.metricRelabelings`

Type

`array`

`.spec.pxc.serviceMonitor.metricRelabelings[]`

Description

RelabelConfig allows dynamic rewriting of the label set, being applied to samples before ingestion. It defines `<metric_relabel_configs>`-section of Prometheus configuration. More info: https://prometheus.io/docs/prometheus/latest/configuration/configuration/#metric_relabel_config

Type

object

Property	Type	Description
<code>action</code>	<code>string</code>	Action to perform based on regex matching. Default is 'replace'
<code>modulus</code>	<code>integer</code>	Modulus to take of the hash of the source label values.
<code>regex</code>	<code>string</code>	Regular expression against which the extracted value is matched. Default is '(.*)'
<code>replacement</code>	<code>string</code>	Replacement value against which a regex replace is performed if the regular expression matches. Regex capture groups are available. Default is '\$1'
<code>separator</code>	<code>string</code>	Separator placed between concatenated source label values. default is ','.
<code>sourceLabels</code>	<code>array</code>	The source labels select values from existing labels. Their content is concatenated using the configured separator and matched against the configured regular expression for the replace, keep, and drop actions.

Property	Type	Description
<code>targetLabel</code>	<code>string</code>	Label to which the resulting value is written in a replace action. It is mandatory for replace actions. Regex capture groups are available.

`.spec.pxc.serviceMonitor.metricRelabelings[].sourceLabels`

Description

The source labels select values from existing labels. Their content is concatenated using the configured separator and matched against the configured regular expression for the replace, keep, and drop actions.

Type

`array`

`.spec.pxc.serviceMonitor.metricRelabelings[].sourceLabels[]`

Type

`string`

`.spec.pxc.slowsql`

Type

`object`

Property	Type	Description
<code>CKTarget</code>	<code>string</code>	

Property	Type	Description
<code>containerSecurityContext</code>	<code>object</code>	SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.
<code>enabled</code>	<code>boolean</code>	
<code>image</code>	<code>string</code>	
<code>imagePullPolicy</code>	<code>string</code>	PullPolicy describes a policy for if/when to pull a container image
<code>logLevel</code>	<code>string</code>	
<code>pmmServer</code>	<code>object</code>	
<code>resources</code>	<code>object</code>	

`.spec.pxc.slowsql.containerSecurityContext`

Description

SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.

Type

`object`

Property	Type	Description
<code>allowPrivilegeEscalation</code>	<code>boolean</code>	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the <code>no_new_privs</code> flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: 1. run as Privileged 2. has CAP_SYS_ADMIN Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>appArmorProfile</code>	<code>object</code>	<code>appArmorProfile</code> is the AppArmor options to use by this container. If set, this profile overrides the pod's <code>appArmorProfile</code>. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>capabilities</code>	<code>object</code>	The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>privileged</code>	<code>boolean</code>	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when <code>spec.os.name</code> is windows.

Property	Type	Description
<code>procMount</code>	<code>string</code>	<code>procMount</code> denotes the type of proc mount to use for the containers. The default value is <code>Default</code> which uses the container runtime defaults for readonly paths and masked paths. This requires the <code>ProcMountType</code> feature flag to be enabled. Note that this field cannot be set when <code>spec.os.name</code> is <code>windows</code>.
<code>readOnlyRootFilesystem</code>	<code>boolean</code>	Whether this container has a read-only root filesystem. Default is <code>false</code>. Note that this field cannot be set when <code>spec.os.name</code> is <code>windows</code>.
<code>runAsGroup</code>	<code>integer</code>	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in <code>PodSecurityContext</code>. If set in both <code>SecurityContext</code> and <code>PodSecurityContext</code>, the value specified in <code>SecurityContext</code> takes precedence. Note that this field cannot be set when <code>spec.os.name</code> is <code>windows</code>.
<code>runAsNonRoot</code>	<code>boolean</code>	Indicates that the container must run as a non-root user. If <code>true</code>, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or <code>false</code>, no such validation will be performed. May also be set in <code>PodSecurityContext</code>. If set in

Property	Type	Description
		both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seccompProfile	object	The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
<code>windowsOptions</code>	<code>object</code>	The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

`.spec.pxc.slowsql.containerSecurityContext.appArmorProfile`

Description

`appArmorProfile` is the AppArmor options to use by this container. If set, this profile overrides the pod's `appArmorProfile`. Note that this field cannot be set when `spec.os.name` is windows.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if <code>type</code> is "Localhost".

Property	Type	Description
<code>type</code>	<code>string</code>	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

`.spec.pxc.slowsql.containerSecurityContext.capabilities`

Description

The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when `spec.os.name` is windows.

Type

`object`

Property	Type	Description
<code>add</code>	<code>array</code>	Added capabilities
<code>drop</code>	<code>array</code>	Removed capabilities

`.spec.pxc.slowsql.containerSecurityContext.capabilities.add`

Description

Added capabilities

Type

`array`

.spec.pxc.slowsql.containerSecurityContext.capabilities.add[]

Description

Capability represent POSIX capabilities type

Type

`string`

.spec.pxc.slowsql.containerSecurityContext.capabilities.drop

Description

Removed capabilities

Type

`array`

.spec.pxc.slowsql.containerSecurityContext.capabilities.drop[]

Description

Capability represent POSIX capabilities type

Type

`string`

.spec.pxc.slowsql.containerSecurityContext.seLinuxOptions

Description

The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Type

object

Property	Type	Description
level	string	Level is SELinux level label that applies to the container.
role	string	Role is a SELinux role label that applies to the container.
type	string	Type is a SELinux type label that applies to the container.
user	string	User is a SELinux user label that applies to the container.

.spec.pxc.slowsql.containerSecurityContext.seccompProfile

Description

The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.

Type

object

Required

type

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.
type	string	type indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied.

.spec.pxc.slowsql.containerSecurityContext.windowsOptions

Description

The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

.spec.pxc.slowsql.pmmServer

Type

object

Property	Type	Description
name	string	
namespace	string	

.spec.pxc.slowsql.resources

Type

object

Property	Type	Description
limits	object	
requests	object	

.spec.pxc.slowsql.resources.limits

Type

object

Property	Type	Description
cpu	string	
ephemeral-storage	string	
memory	string	

.spec.pxc.slowsql.resources.requests

Type

object

Property	Type	Description
cpu	string	
ephemeral-storage	string	
memory	string	

.spec.pxc.tls

Type

object

Property	Type	Description
SANs	array	
issuerConf	object	ObjectReference is a reference to an object with a given name, kind and group.

.spec.pxc.tls.SANs

Type

array

.spec.pxc.tls.SANs[]

Type

`string`

.spec.pxc.tls.issuerConf

Description

ObjectReference is a reference to an object with a given name, kind and group.

Type

`object`

Required

`name`

Property	Type	Description
<code>group</code>	<code>string</code>	Group of the resource being referred to.
<code>kind</code>	<code>string</code>	Kind of the resource being referred to.
<code>name</code>	<code>string</code>	Name of the resource being referred to.

.spec.pxc.upgradeOptions

Type

`object`

Property	Type	Description
<code>apply</code>	<code>string</code>	
<code>schedule</code>	<code>string</code>	

Property	Type	Description
<code>versionServiceEndpoint</code>	<code>string</code>	

.spec.s3Config

Description

S3Config is the configuration of S3 for PXC backup

Type

`object`

.spec.upgradeOption

Description

UpgradeOption is the option to upgrade the MySQL cluster

Type

`object`

Property	Type	Description
<code>autoUpgrade</code>	<code>boolean</code>	AutoUpgrade is the flag to auto upgrade the MySQL
<code>crVersion</code>	<code>string</code>	CRVersion is the version of the CR

.status

Description

MysqlStatus defines the observed state of Mysql

Type

`object`

Property	Type	Description
<code>message</code>	<code>string</code>	Message is the message of the MySQL cluster
<code>mgr</code>	<code>object</code>	MGR is the status of MGR
<code>pxc</code>	<code>object</code>	INSERT ADDITIONAL STATUS FIELD - define observed state of cluster Important: Run "make" to regenerate code after modifying this file PXC is the status of PXC
<code>state</code>	<code>string</code>	State is the state of the MySQL cluster

.status.mgr

Description

MGR is the status of MGR

Type

`object`

Property	Type	Description
<code>clusterSetInitialized</code>	<code>boolean</code>	Indicate if this cluster has been initialized as a member of innodb clusterset
<code>clusterSetRole</code>	<code>string</code>	Indicate the innodb cluster is primary or replica cluster as a member of innodb clusterset

Property	Type	Description
<code>conditions</code>	<code>array</code>	Conditions represents the latest available observations of a MySQLCluster's current state.
<code>currentVersion</code>	<code>string</code>	CurrentVersion represents the current version of the MySQL.
<code>initVersion</code>	<code>string</code>	InitVersion represents the version of the MySQL when it was created.
<code>operatorVersion</code>	<code>string</code>	OperatorVersion means the operator version
<code>primary</code>	<code>array</code>	Primary represents the primary member of the MySQL.
<code>primaryMode</code>	<code>string</code>	PrimaryMode represents the primary mode of the MySQL.
<code>readySize</code>	<code>integer</code>	ReadySize represents the number of members online.
<code>secondary</code>	<code>array</code>	Secondary represents the secondary members of the MySQL.

Property	Type	Description
<code>state</code>	<code>string</code>	State represents the current state of the MySQLCluster.
<code>task</code>	<code>string</code>	Task represents the current task of the MySQLCluster.
<code>upgradeStatus</code>	<code>object</code>	UpgradeStatus represents the upgrade status of the MySQL.

.status.mgr.conditions

Description

Conditions represents the latest available observations of a MySQLCluster's current state.

Type

`array`

.status.mgr.conditions[]

Description

PodCondition contains details for the current condition of this pod.

Type

`object`

Required

`status`

`type`

Property	Type	Description
<code>lastProbeTime</code>	<code>string</code>	Last time we probed the condition.
<code>lastTransitionTime</code>	<code>string</code>	Last time the condition transitioned from one status to another.
<code>message</code>	<code>string</code>	Human-readable message indicating details about last transition.
<code>reason</code>	<code>string</code>	Unique, one-word, CamelCase reason for the condition's last transition.
<code>status</code>	<code>string</code>	Status is the status of the condition. Can be True, False, Unknown. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#pod-conditions ↗
<code>type</code>	<code>string</code>	Type is the type of the condition. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#pod-conditions ↗

.status.mgr.primary

Description

Primary represents the primary member of the MySQL.

Type

`array`

.status.mgr.primary[]

Type

`string`

.status.mgr.secondary

Description

Secondary represents the secondary members of the MySQL.

Type

`array`

.status.mgr.secondary[]

Type

`string`

.status.mgr.upgradeStatus

Description

UpgradeStatus represents the upgrade status of the MySQL.

Type

`object`

Property	Type	Description
<code>crVersion</code>	<code>string</code>	CRVersion means the version of the CR

.status.pxc

Description

INSERT ADDITIONAL STATUS FIELD - define observed state of cluster Important: Run "make" to regenerate code after modifying this file PXC is the status of PXC

Type

object

Property	Type	Description
backup	object	
conditions	array	
haproxy	object	
host	string	
logcollector	object	
message	array	
observedGeneration	integer	
operatorVersion	string	
proxysql	object	
pxc	object	
state	string	

.status.pxc.backup

Type

object

Property	Type	Description
image	string	

Property	Type	Description
message	string	
ready	integer	
size	integer	
status	string	
version	string	

.status.pxc.conditions

Type

array

.status.pxc.conditions[]

Type

object

Property	Type	Description
lastTransitionTime	string	
message	string	
reason	string	
status	string	
type	string	

.status.pxc.haproxy

Type

`object`

Property	Type	Description
<code>image</code>	<code>string</code>	
<code>message</code>	<code>string</code>	
<code>ready</code>	<code>integer</code>	
<code>size</code>	<code>integer</code>	
<code>status</code>	<code>string</code>	
<code>version</code>	<code>string</code>	

`.status.pxc.logcollector`

Type

`object`

Property	Type	Description
<code>image</code>	<code>string</code>	
<code>message</code>	<code>string</code>	
<code>ready</code>	<code>integer</code>	
<code>size</code>	<code>integer</code>	
<code>status</code>	<code>string</code>	
<code>version</code>	<code>string</code>	

`.status.pxc.message`

Type

`array`

`.status.pxc.message[]`

Type

`string`

`.status.pxc.proxysql`

Type

`object`

Property	Type	Description
<code>image</code>	<code>string</code>	
<code>message</code>	<code>string</code>	
<code>ready</code>	<code>integer</code>	
<code>size</code>	<code>integer</code>	
<code>status</code>	<code>string</code>	
<code>version</code>	<code>string</code>	

`.status.pxc.pxc`

Type

`object`

Property	Type	Description
<code>image</code>	<code>string</code>	
<code>message</code>	<code>string</code>	

Property	Type	Description
ready	integer	
size	integer	
status	string	
version	string	

API Endpoints

The following API endpoints are available:

- `/apis/middleware.alauda.io/v1/namespaces/{namespace}/mysqls`
 - `DELETE` : delete collection of Mysql
 - `GET` : list objects of kind Mysql
 - `POST` : create a new Mysql
- `/apis/middleware.alauda.io/v1/namespaces/{namespace}/mysqls/{name}`
 - `DELETE` : delete the specified Mysql
 - `GET` : read the specified Mysql
 - `PATCH` : partially update the specified Mysql
 - `PUT` : replace the specified Mysql
- `/apis/middleware.alauda.io/v1/namespaces/{namespace}/mysqls/{name}/status`
 - `GET` : read status of the specified Mysql
 - `PATCH` : partially update status of the specified Mysql
 - `PUT` : replace status of the specified Mysql

`/apis/middleware.alauda.io/v1/namespaces/{namespace}/mysqls`

HTTP method

DELETE

Description

delete collection of Mysql

HTTP responses

HTTP code	Response body
200 - OK	<code>Status</code> schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind Mysql

HTTP responses

HTTP code	Response body
200 - OK	<code>MysqlList</code> schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new Mysql

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing

Parameter	Type	Description
		of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>Mysql</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>Mysql</code> schema
201 - Created	<code>Mysql</code> schema
202 - Accepted	<code>Mysql</code> schema
401 - Unauthorized	Empty

/apis/middleware.alauda.io/v1/namespaces/{namespace}/mysqls/{name}

HTTP method

DELETE

Description

delete the specified Mysql

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	<code>Status</code> ✓ schema
202 - Accepted	<code>Status</code> ✓ schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified Mysql

HTTP responses

HTTP code	Response body
200 - OK	<code>Mysql</code> schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified Mysql

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>Mysql</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified Mysql

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>Mysql</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>Mysql</code> schema
201 - Created	<code>Mysql</code> schema
401 - Unauthorized	Empty

/apis/middleware.alauda.io/v1/namespaces/{namespace}/mysqls/{name}/status

HTTP method

`GET`

Description

read status of the specified Mysql

HTTP responses

HTTP code	Response body
200 - OK	<code>Mysql</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified Mysql

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>Mysql</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified `Mysql`

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>Mysql</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>Mysql</code> schema
201 - Created	<code>Mysql</code> schema
401 - Unauthorized	Empty



MGR Backup

Description

MySQLBackup is the Schema for the mysqlbackups API

Type

object

Specification

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗

Property	Type	Description
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	MySQLBackupSpec defines the desired state of MySQLBackup
status	object	MySQLBackupStatus defines the observed state of MySQLBackup

.spec

Description

MySQLBackupSpec defines the desired state of MySQLBackup

Type

object

Property	Type	Description
cluster	object	Cluster is the name of source cluster.

Property	Type	Description
<code>fullExecutor</code>	<code>string</code>	Executor is the executor of full backup.
<code>mysqlShellDump</code>	<code>object</code>	DumpOption is the dump option for backup.
<code>storage</code>	<code>object</code>	Storage is the storage information of backup for.
<code>storeMeta</code>	<code>boolean</code>	StoreMeta is the flag of store meta data.
<code>type</code>	<code>string</code>	Type means full or increment backup.

.spec.cluster

Description

Cluster is the name of source cluster.

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

.spec.mysqlShellDump

Description

DumpOption is the dump option for backup.

Type

object

Property	Type	Description
<code>excludeSchemas</code>	array	ExcludeSchemas specifies the schemas to exclude from the backup.
<code>excludeTables</code>	array	ExcludeTables specifies the tables to exclude from the backup, using the 'schema.table' format.
<code>includeSchemas</code>	array	IncludeSchemas specifies the schemas to include in the backup.
<code>includeTables</code>	array	IncludeTables specifies the tables to include in the backup, using the 'schema.table' format.
<code>maxRate</code>	integer	MaxRate specifies the maximum rate at which data is transferred during the backup operation.
<code>threads</code>	integer	Threads specifies the number of threads to use for the backup operation.

`.spec.mysqlShellDump.excludeSchemas`

Description

ExcludeSchemas specifies the schemas to exclude from the backup.

Type

array

`.spec.mysqlShellDump.excludeSchemas[]`

Type

string

`.spec.mysqlShellDump.excludeTables`

Description

ExcludeTables specifies the tables to exclude from the backup, using the 'schema.table' format.

Type

array

`.spec.mysqlShellDump.excludeTables[]`

Type

string

`.spec.mysqlShellDump.includeSchemas`

Description

IncludeSchemas specifies the schemas to include in the backup.

Type

array

.spec.mysqlShellDump.includeSchemas[]

Type

string

.spec.mysqlShellDump.includeTables

Description

IncludeTables specifies the tables to include in the backup, using the 'schema.table' format.

Type

array

.spec.mysqlShellDump.includeTables[]

Type

string

.spec.storage

Description

Storage is the storage information of backup for.

Type

object

Property	Type	Description
s3	object	S3 means s3 compatible object storage

.spec.storage.s3

Description

S3 means s3 compatible object storage

Type

object

Property	Type	Description
bucket	string	Bucket in which to store the Backup.
endpoint	string	Endpoint (hostname only or fully qualified URI) of S3 compatible storage service.
region	string	Region in which the S3 compatible bucket is located.
secret	object	Secret is a reference to the Secret containing the credentials authenticating with the S3 compatible storage service.

.spec.storage.s3.secret

Description

Secret is a reference to the Secret containing the credentials authenticating with the S3 compatible storage service.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More

Property	Type	Description
		info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

.status

Description

MySQLBackupStatus defines the observed state of MySQLBackup

Type

object

Property	Type	Description
<code>allocated</code>	<code>string</code>	Allocated is the name of pod allocate to.
<code>dataEndTime</code>	<code>string</code>	DataEndTime is the stop time of increment backup or full backup data.
<code>dataStartTime</code>	<code>string</code>	DataStartTime is the start time of increment backup data.
<code>executor</code>	<code>string</code>	Executor is the executor of backup.
<code>finishTime</code>	<code>string</code>	FinishTime is the finish time of backup runs.

Property	Type	Description
<code>gtidEnd</code>	<code>string</code>	GtidEnd is the stop gtid of backup data
<code>gtidFullPrevious</code>	<code>string</code>	FullGtid is the gtid of full backup if skip a full backup.
<code>gtidPrevious</code>	<code>string</code>	GtidPrevious is the gtid of previous backup.
<code>gtidStart</code>	<code>string</code>	GtidStart is the start gtid of backup data
<code>mark</code>	<code>string</code>	Mark is the mark of backup
<code>memberSize</code>	<code>integer</code>	MemberSize is the size of cluster member.
<code>message</code>	<code>string</code>	Message is the message of run result of backup
<code>path</code>	<code>string</code>	Path is the path of the file stored in storage.
<code>startTime</code>	<code>string</code>	StartTime is the start time of backup runs.
<code>state</code>	<code>string</code>	State is the backup state and enumeration of success, fail or running.

Property	Type	Description
<code>storeMeta</code>	<code>boolean</code>	StoreMeta is the flag of store meta CR.

API Endpoints

The following API endpoints are available:

- `/apis/mysql.middleware.alauda.io/v1/namespaces/{namespace}/mysqlbackups`
 - `DELETE` : delete collection of MySQLBackup
 - `GET` : list objects of kind MySQLBackup
 - `POST` : create a new MySQLBackup
- `/apis/mysql.middleware.alauda.io/v1/namespaces/{namespace}/mysqlbackups/{name}`
 - `DELETE` : delete the specified MySQLBackup
 - `GET` : read the specified MySQLBackup
 - `PATCH` : partially update the specified MySQLBackup
 - `PUT` : replace the specified MySQLBackup
- `/apis/mysql.middleware.alauda.io/v1/namespaces/{namespace}/mysqlbackups/{name}/status`
 - `GET` : read status of the specified MySQLBackup
 - `PATCH` : partially update status of the specified MySQLBackup
 - `PUT` : replace status of the specified MySQLBackup

`/apis/mysql.middleware.alauda.io/v1/namespaces/{namespace}/mysqlbackups`

HTTP method

DELETE

Description

delete collection of MySQLBackup

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind MySQLBackup

HTTP responses

HTTP code	Response body
200 - OK	MySQLBackupList schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new MySQLBackup

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing

Parameter	Type	Description
		of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>MySQLBackup</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>MySQLBackup</code> schema
201 - Created	<code>MySQLBackup</code> schema
202 - Accepted	<code>MySQLBackup</code> schema
401 - Unauthorized	Empty

/apis/mysql.middleware.alauda.io/v1/namespaces/{namespace}/mysqlbackups/{name}

HTTP method

DELETE

Description

delete the specified MySQLBackup

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	<code>Status</code> ✓ schema
202 - Accepted	<code>Status</code> ✓ schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified MySQLBackup

HTTP responses

HTTP code	Response body
200 - OK	<code>MySQLBackup</code> schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified MySQLBackup

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>MySQLBackup</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified MySQLBackup

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	MySQLBackup schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	MySQLBackup schema
201 - Created	MySQLBackup schema
401 - Unauthorized	Empty

/apis/mysql.middleware.alauda.io/v1/namespaces/{namespace}/mysqlbackups/{name}/status

HTTP method

GET

Description

read status of the specified MySQLBackup

HTTP responses

HTTP code	Response body
200 - OK	MySQLBackup schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified MySQLBackup

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>MySQLBackup</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified `MySQLBackup`

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>MySQLBackup</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>MySQLBackup</code> schema
201 - Created	<code>MySQLBackup</code> schema
401 - Unauthorized	Empty

MGR Restore

Description

MySQLRestore is the Schema for the MySQLRestores API

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗

Property	Type	Description
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	MySQLRestoreSpec defines the desired state of MySQLRestore
status	object	MySQLRestoreStatus defines the observed state of MySQLRestore

.spec

Description

MySQLRestoreSpec defines the desired state of MySQLRestore

Type

object

Property	Type	Description
mgr	object	MGR use to create mgr restore schedule

Property	Type	Description
<code>mysql</code>	<code>object</code>	MySQL use to create a new MySQL instance

.spec.mgr

Description

MGR use to create mgr restore schedule

Type

`object`

Property	Type	Description
<code>backup</code>	<code>object</code>	Backup is the name of Backup CR you want to restore.
<code>cluster</code>	<code>object</code>	Cluster is the cluster where the data will be restored to.
<code>mysqlShellLoad</code>	<code>object</code>	LoadOption is the load option for restore
<code>source</code>	<code>object</code>	Source is the cluster where the data comes form
<code>timePoint</code>	<code>string</code>	TimePoint is the deadline of data you want to restore.

.spec.mgr.backup

Description

Backup is the name of Backup CR you want to restore.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

.spec.mgr.cluster

Description

Cluster is the cluster where the data will be restored to.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

.spec.mgr.mysqlShellLoad

Description

LoadOption is the load option for restore

Type

object

Property	Type	Description
<code>excludeUsers</code>	<code>array</code>	ExcludeUsers specifies the users to exclude from the backup.
<code>includeSchemas</code>	<code>array</code>	IncludeSchemas specifies the schemas to include in the backup.
<code>includeTables</code>	<code>array</code>	IncludeTables specifies the tables to include in the backup, using the 'schema.table' format.
<code>includeUsers</code>	<code>array</code>	IncludeUsers specifies the users to include in the backup.
<code>loadUsers</code>	<code>boolean</code>	loadUsers specifies whether to load users during the restore operation.
<code>maxBytesPerTransaction</code>	<code>integer</code>	MaxBytesPerTransaction specifies the maximum number of bytes to use per transaction during the restore operation.
<code>threads</code>	<code>integer</code>	Threads specifies the number of threads to use for the restore operation.

.spec.mgr.mysqlShellLoad.excludeUsers

Description

ExcludeUsers specifies the users to exclude from the backup.

Type

array

.spec.mgr.mysqlShellLoad.excludeUsers[]

Type

string

.spec.mgr.mysqlShellLoad.includeSchemas

Description

IncludeSchemas specifies the schemas to include in the backup.

Type

array

.spec.mgr.mysqlShellLoad.includeSchemas[]

Type

string

.spec.mgr.mysqlShellLoad.includeTables

Description

IncludeTables specifies the tables to include in the backup, using the 'schema.table' format.

Type

array

`.spec.mgr.mysqlShellLoad.includeTables[]`

Type

`string`

`.spec.mgr.mysqlShellLoad.includeUsers`

Description

IncludeUsers specifies the users to include in the backup.

Type

`array`

`.spec.mgr.mysqlShellLoad.includeUsers[]`

Type

`string`

`.spec.mgr.source`

Description

Source is the cluster where the data comes form

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	
<code>namespace</code>	<code>string</code>	
<code>type</code>	<code>string</code>	

`.spec.mysql`

Description

MySQL use to create a new MySQL instance

Type

object

Required

version

Property	Type	Description
crVersion	string	Deprecated: use upgradeOption.CrVersion
ipFamilyPrefer	string	IPFamilyPrefer is the preferred IP family
mgr	object	MGR use to configure MySQLCluster
params	object	Params is the configuration of MySQL Server
paras	object	Paras deprecated: this field is deprecated, please use spec.params instead
password	string	Password Deprecated: this field is deprecated
pause	boolean	Pause is the flag to pause the MySQL cluster

Property	Type	Description
<code>pxc</code>	<code>object</code>	PXC use to configure PerconaXtraDBCluster
<code>runAsRoot</code>	<code>boolean</code>	RunAsRoot is the flag to run MySQL Server as root
<code>s3Config</code>	<code>object</code>	S3Config is the configuration of S3 for PXC backup
<code>upgradeOption</code>	<code>object</code>	UpgradeOption is the option to upgrade the MySQL cluster
<code>useSafeConf</code>	<code>boolean</code>	UseSafeConf is the flag to use safe configuration
<code>version</code>	<code>string</code>	Version is the version of the MySQL Server

.spec.mysql.mgr

Description

MGR use to configure MySQLCluster

Type

`object`

Property	Type	Description
affinity	object	If specified, affinity will define the pod's scheduling constraints
backupVolumeClaimTemplate	object	BackupVolumeClaimTemplate allows a user to specify a volume to temporarily store the data for a backup before it being shipped to object storage.
baseServerId	integer	BaseServerID defines the base number used to generate a unique server_id for MySQL instances in the cluster. Valid range 1 to 4294967286. If omitted in the manifest file (or set to 0) defaultBaseServerID value will be used.
certConfig	object	CertConfig allows a user to specify custom CA certificates with advanced option.
config	object	Config allows a user to specify a custom configuration file for MySQL.
configuration	string	Mysql configuration
enableStorage	boolean	EnableStorage allow a user to enable mysql storage.
image	object	Image allows a user to specify image.

Property	Type	Description
<code>imagePullPolicy</code>	<code>string</code>	PullPolicy describes a policy for if/when to pull a container image
<code>imagePullSecret</code>	<code>array</code>	ImagePullSecret defines the name of the secret that contains the required credentials for pulling from specified Repository.
<code>ipFamilyPolicy</code>	<code>string</code>	IpFamilyPolicy means the IP family policy for cluster
<code>ipFamilyPreference</code>	<code>string</code>	IpFamilyPreference means the preferred IP family for cluster
<code>jemallocConf</code>	<code>string</code>	JemallocConf means the jemalloc configuration
<code>majorViewIDQuorum</code>	<code>boolean</code>	MajorViewIDQuorum means use the major view ID to determine the quorum
<code>members</code>	<code>integer</code>	Members defines the number of MySQL instance in the cluster. InnoDB clustering supports between 1-9 members.
<code>monitor</code>	<code>object</code>	Monitor defines MySQL monitor specification

Property	Type	Description
<code>multiClusterReplication</code>	<code>object</code>	Config this field to support multi cluster replication disaster recovery.
<code>multiMaster</code>	<code>boolean</code>	MultiMaster defines the mode of the MySQL cluster. If set to true, all instances will be R/W. If false (the default), only a single instance will be R/W and the rest will be R/O.
<code>nodeSelector</code>	<code>object</code>	NodeSelector is a selector which must be true for a pod to fit on a node. Selector which must match a node's labels for the pod to be scheduled on that node. For more info: https://kubernetes.io/docs/concepts/configuration/node-selector/
<code>paras</code>	<code>object</code>	Paras provide paras template feature
<code>patches</code>	<code>object</code>	Patches use to apply patches to the MySQL container
<code>pause</code>	<code>boolean</code>	Pause means pause the cluster
<code>podCIDR</code>	<code>string</code>	PodCIDR allows a user to specify custom CIDR.

Property	Type	Description
podManagementPolicy	string	PodManagementPolicy defines the policy for creating pods under the statefulset
readOnlyRootFilesystem	boolean	ReadOnlyRootFilesystem means the root filesystem is read-only
repository	string	Repository defines the image repository from which to pull the MySQL server image.
resources	object	Resources holds ResourceRequirements for the Agent & Server Containers
rootPasswordSecret	object	If defined, we use this secret for configuring the MYSQL_ROOT_PASSWORD. If it is not set, we generate a secret dynamically.
router	object	Router defines the MySQL Router
runAsRoot	boolean	RunAsRoot defines whether the MySQL container should run as root.
securityContext	object	SecurityContext holds Pod-level security attributes and common Container settings.

Property	Type	Description
<code>slowSQL</code>	<code>object</code>	SlowSQL define MySQL slow sql spec
<code>sslSecret</code>	<code>object</code>	SSLSecret allows a user to specify custom CA certificate, server certificate and server key for gr replication SSL.
<code>strictSecurityModeEnabled</code>	<code>boolean</code>	Enable this option to prevent sensitive information appearing in plain text in environment variables o a container.
<code>tolerations</code>	<code>array</code>	Tolerations allows specifying a list of tolerations fi controlling which set of Nodes a Pod can be sche on
<code>upgradeOption</code>	<code>object</code>	UpgradeOption defines the upgrade option for the MySQL cluster.
<code>useJemalloc</code>	<code>boolean</code>	UseJemalloc means use jemalloc for MySQL
<code>version</code>	<code>string</code>	Version defines the MySQL Docker image version
<code>volumeClaimTemplate</code>	<code>object</code>	VolumeClaimTemplate allows a user to specify ho volumes inside a MySQL cluster

.spec.mysql.mgr.affinity

Description

If specified, affinity will define the pod's scheduling constraints

Type

object

Property	Type	Description
<code>nodeAffinity</code>	object	Describes node affinity scheduling rules for the pod.
<code>podAffinity</code>	object	Describes pod affinity scheduling rules (e.g. co-locate this pod in the same node, zone, etc. as some other pod(s)).
<code>podAntiAffinity</code>	object	Describes pod anti-affinity scheduling rules (e.g. avoid putting this pod in the same node, zone, etc. as some other pod(s)).

.spec.mysql.mgr.affinity.nodeAffinity

Description

Describes node affinity scheduling rules for the pod.

Type

object

Property	Type	Description
<code>preferredDuringSchedulingIgnoredDuringExecution</code>	array	The scheduler will prefer to schedule pods to

Property	Type	Description
		nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution affinity expressions, etc.) compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>object</code>	If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some

Property	Type	Description
		point during pod execution (e.g. due to an update), the system may or may not try to eventually evict the pod from its node.

`.spec.mysql.mgr.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest sum are the most preferred.

Type

array

`.spec.mysql.mgr.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

An empty preferred scheduling term matches all objects with implicit weight 0 (i.e. it's a no-op). A null preferred scheduling term matches no objects (i.e. is also a no-op).

Type

object

Required

preference

weight

Property	Type	Description
preference	object	A node selector term, associated with the corresponding weight.
weight	integer	Weight associated with matching the corresponding nodeSelectorTerm, in the range 1-100.

.spec.mysql.mgr.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference

Description

A node selector term, associated with the corresponding weight.

Type

object

Property	Type	Description
matchExpressions	array	A list of node selector requirements by node's labels.
matchFields	array	A list of node selector requirements by node's fields.

.spec.mysql.mgr.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions

Description

A list of node selector requirements by node's labels.

Type

array

.spec.mysql.mgr.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[]

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.

Property	Type	Description
<code>values</code>	<code>array</code>	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

`.spec.mysql.mgr.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values`

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.mysql.mgr.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values[]`

Type

`string`

`.spec.mysql.mgr.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields`

Description

A list of node selector requirements by node's fields.

Type

array

`.spec.mysql.mgr.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[]`

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
<code>key</code>	<code>string</code>	The label key that the selector applies to.
<code>operator</code>	<code>string</code>	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
<code>values</code>	<code>array</code>	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.mysql.mgr.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.mgr.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values[]

Type

string

.spec.mysql.mgr.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to an update), the system may or may not try to eventually evict the pod from its node.

Type

object

Required

nodeSelectorTerms

Property	Type	Description
<code>nodeSelectorTerms</code>	<code>array</code>	Required. A list of node selector terms. The terms are ORed.

`.spec.mysql.mgr.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms`

Description

Required. A list of node selector terms. The terms are ORed.

Type

`array`

`.spec.mysql.mgr.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[]`

Description

A null or empty node selector term matches no objects. The requirements of them are ANDed. The TopologySelectorTerm type implements a subset of the NodeSelectorTerm.

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	A list of node selector requirements by node's labels.
<code>matchFields</code>	<code>array</code>	A list of node selector requirements by node's fields.

.spec.mysql.mgr.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions

Description

A list of node selector requirements by node's labels.

Type

array

.spec.mysql.mgr.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[]

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.

Property	Type	Description
<code>values</code>	<code>array</code>	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.mysql.mgr.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

`array`

.spec.mysql.mgr.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values[]

Type

`string`

.spec.mysql.mgr.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values[]

hFields

Description

A list of node selector requirements by node's fields.

Type

array

.spec.mysql.mgr.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[]

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will

Property	Type	Description
		be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.mysql.mgr.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.mgr.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values[]

Type

string

.spec.mysql.mgr.affinity.podAffinity

Description

Describes pod affinity scheduling rules (e.g. co-locate this pod in the same node, zone, etc. as some other pod(s)).

Type

object

Property	Type	Description
<code>preferredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, <code>requiredDuringSchedulingIgnoredDuringExecution</code>, affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding <code>podAffinityTerm</code>; the node(s) with the highest sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	If the affinity requirements specified by this field are not met at scheduling time, the pod will not be

Property	Type	Description
		scheduled onto the node. If the affinity requirement: specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

.spec.mysql.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

array

`.spec.mysql.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

The weights of all of the matched `WeightedPodAffinityTerm` fields are added per-node to find the most preferred node(s)

Type

`object`

Required

`podAffinityTerm``weight`

Property	Type	Description
<code>podAffinityTerm</code>	<code>object</code>	Required. A pod affinity term, associated with the corresponding weight.
<code>weight</code>	<code>integer</code>	weight associated with matching the corresponding <code>podAffinityTerm</code> , in the range 1-100.

`.spec.mysql.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm`

Description

Required. A pod affinity term, associated with the corresponding weight.

Type

`object`

Required

`topologyKey`

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field

Property	Type	Description
		and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.mysql.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSele`

ctor

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.mysql.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.mysql.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSele

ctor.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mysql.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This

array is replaced during a strategic merge patch.

Type

array

.spec.mysql.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]

Type

string

.spec.mysql.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mysql.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti)

affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.mysql.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]

Type

string

.spec.mysql.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.mysql.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatch

LabelKeys[]

Type

string

.spec.mysql.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.mysql.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespa

ceSelector.matchExpressions

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.mysql.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.

Property	Type	Description
<code>values</code>	<code>array</code>	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.mysql.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.mysql.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]`

Type

`string`

`.spec.mysql.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mysql.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.mysql.mgr.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

string

.spec.mysql.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may

or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Type

array

.spec.mysql.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running

Type

object

Required

topologyKey

Property	Type	Description
labelSelector	object	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
matchLabelKeys	array	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity.

Property	Type	Description
		Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Property	Type	Description
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.mysql.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	matchExpressions is a list of label selector requirements. The requirements are ANDed.

Property	Type	Description
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.mysql.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

`key`

`operator`

Property	Type	Description
<code>key</code>	<code>string</code>	key is the label key that the selector applies to.
<code>operator</code>	<code>string</code>	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
<code>values</code>	<code>array</code>	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.mysql.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.mysql.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]`

Type

`string`

`.spec.mysql.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

`object`

`.spec.mysql.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys`

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

`array`

`.spec.mysql.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]`

Type

`string`

`.spec.mysql.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys`

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

`.spec.mysql.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]`

Type

string

`.spec.mysql.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.mysql.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mysql.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.match

Expressions[].values[]

Type

string

.spec.mysql.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mysql.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.mysql.mgr.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]

Type

string

.spec.mysql.mgr.affinity.podAntiAffinity

Description

Describes pod anti-affinity scheduling rules (e.g. avoid putting this pod in the same node, zone, etc. as some other pod(s)).

Type

object

Property	Type	Description
<code>preferredDuringSchedulingIgnoredDuringExecution</code>	array	The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, <code>requiredDuringSchedulingIgnoredDuringExecution</code> , anti-affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding <code>podAffinityTerm</code> ; the

Property	Type	Description
		node(s) with the highest sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

`.spec.mysql.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the

expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling anti-affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

array

.spec.mysql.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

Description

The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Type

object

Required

podAffinityTerm

weight

Property	Type	Description
<code>podAffinityTerm</code>	<code>object</code>	Required. A pod affinity term, associated with the corresponding weight.
<code>weight</code>	<code>integer</code>	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

.spec.mysql.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm

Description

Required. A pod affinity term, associated with the corresponding weight.

Type

object

Required

topologyKey

Property	Type	Description
labelSelector	object	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
matchLabelKeys	array	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
mismatchLabelKeys	array	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with

Property	Type	Description
		<code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both <code>mismatchLabelKeys</code> and <code>labelSelector</code> . Also, <code>mismatchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. This is a beta field and requires enabling <code>MatchLabelKeysInPodAffinity</code> feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the <code>namespaces</code> field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector (<code>{}</code>) matches all namespaces.
<code>namespaces</code>	<code>array</code>	<code>namespaces</code> specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by <code>namespaceSelector</code> . null or empty namespaces list and null <code>namespaceSelector</code> means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the <code>labelSelector</code> in the specified namespaces, where co-located is defined as running on a node whose value of the label with key <code>topologyKey</code> matches that of any

Property	Type	Description
		node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.mysql.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.mysql.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mysql.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]

Type

string

.spec.mysql.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

`.spec.mysql.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys`

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

`.spec.mysql.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]`

Type

string

`.spec.mysql.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys`

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti)

affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both `mismatchLabelKeys` and `labelSelector`. Also, `mismatchLabelKeys` cannot be set when `labelSelector` isn't set. This is a beta field and requires enabling `MatchLabelKeysInPodAffinity` feature gate (enabled by default).

Type

array

`.spec.mysql.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]`

Type

string

`.spec.mysql.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the `namespaces` field. A null selector and null or empty namespaces list means "this pod's namespace". An empty selector (`{}`) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.

Property	Type	Description
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces.selector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.mysql.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces.selector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

`key`

`operator`

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mysql.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces.selector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces.selector.matchExpressions[].values[]

Type

`string`

.spec.mysql.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

`object`

.spec.mysql.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

`array`

.spec.mysql.mgr.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

`string`

.spec.mysql.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Type

array

.spec.mysql.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running

Type

object

Required

topologyKey

Property	Type	Description
labelSelector	object	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Property	Type	Description
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Property	Type	Description
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector (<code>{}</code>) matches all namespaces.
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.mysql.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.mysql.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mysql.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.mysql.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]`

Type

string

`.spec.mysql.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

`.spec.mysql.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys`

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

`.spec.mysql.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]`

Type

string

`.spec.mysql.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys`

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

`.spec.mysql.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]`

Type

string

`.spec.mysql.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
matchExpressions	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
matchLabels	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.mysql.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.mysql.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

keyoperator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mysql.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]

Type

string

.spec.mysql.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mysql.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.mysql.mgr.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]

Type

string

.spec.mysql.mgr.backupVolumeClaimTemplate

Description

BackupVolumeClaimTemplate allows a user to specify a volume to temporarily store the data for a backup prior to it being shipped to object storage.

Type

object

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

Property	Type	Description
metadata	ObjectMeta ↗	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata ↗
spec	object	spec defines the desired characteristics of a volume requested by a pod author. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims ↗
status	object	status represents the current information/status of a persistent volume claim. Read-only. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims ↗

.spec.mysql.mgr.backupVolumeClaimTemplate.spec

Description

spec defines the desired characteristics of a volume requested by a pod author. More info:
<https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims>

Type

object

Property	Type	Description
<code>accessModes</code>	<code>array</code>	<code>accessModes</code> contains the desired access modes volume should have. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1
<code>dataSource</code>	<code>object</code>	<code>dataSource</code> field can be used to specify either: • An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) • An existing PVC (PersistentVolumeClaim) If the provisioner or an external controller can support the specified data source, it will create a new volume based on the contents of the specified data source. When the AnyVolumeDataSource feature gate is enabled, <code>dataSource</code> contents will be copied to <code>dataSourceRef</code>, and <code>dataSourceRef</code> contents will be copied to <code>dataSource</code> when <code>dataSourceRef.namespace</code> is not specified. If <code>dataSourceRef.namespace</code> is specified, then <code>dataSourceRef</code> will not be copied to <code>dataSource</code>.
<code>dataSourceRef</code>	<code>object</code>	<code>dataSourceRef</code> specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a PersistentVolumeClaim object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replicate the functionality of the <code>dataSource</code> field and as such both fields are non-empty, they must have the same

Property	Type	Description
		value. For backwards compatibility, when namespace isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in dataSourceRef, dataSource isn't set to the same value and must be empty. There are three important differences between dataSource and dataSourceRef: • While dataSource only allows two specific type objects, dataSourceRef allows any non-core objects as well as PersistentVolumeClaim objects. • While dataSource ignores disallowed values (dropping them), dataSourceRef preserves all values, and generates an error if a disallowed value is specified. • While dataSource only allows local objects, dataSourceRef allows objects in any namespace (Beta) Using this field requires the AnyVolumeDataSource feature gate to be enabled (Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

Property	Type	Description
resources	object	resources represents the minimum resources the volume should have. If RecoverVolumeExpansionFailure feature is enable users are allowed to specify resource requirement that are lower than previous value but must still be higher than capacity recorded in the status field of claim. More info: https://kubernetes.io/docs/concepts/storage/persistentvolumes#resources
selector	object	selector is a label query over volumes to consider binding.
storageClassName	string	storageClassName is the name of the StorageClass required by the claim. More info: https://kubernetes.io/docs/concepts/storage/persistentvolumes#class-1
volumeAttributesClassName	string	volumeAttributesClassName may be used to set the VolumeAttributesClass used by this claim. If specified the CSI driver will create or update the volume with attributes defined in the corresponding VolumeAttributesClass. This has a different purpose than storageClassName, it can be changed after the claim is created. An empty string value means that VolumeAttributesClass will be applied to the claim it's not allowed to reset this field to empty string once it is set. If unspecified and the PersistentVolumeClaim is unbound, the default VolumeAttributesClass will be used by the persistentvolume controller if it exists. If the

Property	Type	Description
		resource referred to by volumeAttributesClass does not exist, this PersistentVolumeClaim will be set to Pending state, as reflected by the modifyVolumeStatus field, until such as a resource exists. More info: https://kubernetes.io/docs/concepts/storage/volume-attributes-classes/ (Beta) Using this field requires the VolumeAttributesClass feature gate to be enabled (off by default).
volumeMode	string	volumeMode defines what type of volume is required by the claim. Value of Filesystem is implied when not included in claim spec.
volumeName	string	volumeName is the binding reference to the PersistentVolume backing this claim.

.spec.mysql.mgr.backupVolumeClaimTemplate.spec.accessModes

Description

accessModes contains the desired access modes the volume should have. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1>

Type

array

.spec.mysql.mgr.backupVolumeClaimTemplate.spec.accessModes[]

Type

`string`

`.spec.mysql.mgr.backupVolumeClaimTemplate.spec.dataSource`

Description

`dataSource` field can be used to specify either: * An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) * An existing PVC (PersistentVolumeClaim) If the provisioner or an external controller can support the specified data source, it will create a new volume based on the contents of the specified data source. When the `AnyVolumeDataSource` feature gate is enabled, `dataSource` contents will be copied to `dataSourceRef`, and `dataSourceRef` contents will be copied to `dataSource` when `dataSourceRef.namespace` is not specified. If the namespace is specified, then `dataSourceRef` will not be copied to `dataSource`.

Type

`object`

Required

`kind``name`

Property	Type	Description
<code>apiGroup</code>	<code>string</code>	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
<code>kind</code>	<code>string</code>	Kind is the type of resource being referenced
<code>name</code>	<code>string</code>	Name is the name of resource being referenced

.spec.mysql.mgr.backupVolumeClaimTemplate.spec.dataSourceRef

Description

dataSourceRef specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a PersistentVolumeClaim object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replace the functionality of the dataSource field and as such if both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in dataSourceRef, dataSource isn't set to the same value and must be empty. There are three important differences between dataSource and dataSourceRef: * While dataSource only allows two specific types of objects, dataSourceRef allows any non-core object, as well as PersistentVolumeClaim objects. * While dataSource ignores disallowed values (dropping them), dataSourceRef preserves all values, and generates an error if a disallowed value is specified. * While dataSource only allows local objects, dataSourceRef allows objects in any namespaces. (Beta) Using this field requires the AnyVolumeDataSource feature gate to be enabled. (Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

Type

object

Required

kind

name

Property	Type	Description
apiGroup	string	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.

Property	Type	Description
<code>kind</code>	<code>string</code>	Kind is the type of resource being referenced
<code>name</code>	<code>string</code>	Name is the name of resource being referenced
<code>namespace</code>	<code>string</code>	Namespace is the namespace of resource being referenced Note that when a namespace is specified, a gateway.networking.k8s.io/ReferenceGrant object is required in the referent namespace to allow that namespace's owner to accept the reference. See the ReferenceGrant documentation for details. (Alpha) This field requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

`.spec.mysql.mgr.backupVolumeClaimTemplate.spec.resources`

Description

`resources` represents the minimum resources the volume should have. If RecoverVolumeExpansionFailure feature is enabled users are allowed to specify resource requirements that are lower than previous value but must still be higher than capacity recorded in the status field of the claim. More info:

<https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources>

Type

`object`

Property	Type	Description
<code>limits</code>	<code>object</code>	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗
<code>requests</code>	<code>object</code>	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

`.spec.mysql.mgr.backupVolumeClaimTemplate.spec.resources.limits`

Description

Limits describes the maximum amount of compute resources allowed. More info:
<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.mysql.mgr.backupVolumeClaimTemplate.spec.resources.requests`

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an

implementation-defined value. Requests cannot exceed Limits. More info:
<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

.spec.mysql.mgr.backupVolumeClaimTemplate.spec.selector

Description

selector is a label query over volumes to consider for binding.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.mysql.mgr.backupVolumeClaimTemplate.spec.selector.matchExpressions

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.mysql.mgr.backupVolumeClaimTemplate.spec.selector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.mysql.mgr.backupVolumeClaimTemplate.spec.selector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.mgr.backupVolumeClaimTemplate.spec.selector.matchExpressions[].values[]

Type

string

.spec.mysql.mgr.backupVolumeClaimTemplate.spec.selector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mysql.mgr.backupVolumeClaimTemplate.status

Description

status represents the current information/status of a persistent volume claim. Read-only.
More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims>

Type

object

Property	Type	Description
<code>accessModes</code>	<code>array</code>	accessModes contains the actual access modes of the volume backing the PVC. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1
<code>allocatedResourceStatuses</code>	<code>object</code>	allocatedResourceStatuses stores status of resources being resized for the given PVC. Key names must follow standard Kubernetes label syntax. Valid values are either: * Un-prefixed keys: - storage - the controller is resizing the volume. * Custom resources must use implementation-defined prefixed names such as "example.com/my-custom-resource". Apart from these values - keys that are unprefixed or have the kubernetes.io prefix are considered reserved and hence may not be used. ClaimResourceStatus can be in any of following states: - ControllerResizeInProgress: State set when resize controller starts resizing the volume. - ControllerResizeFailed: State set when resize controller has failed in resize controller with a termination error. - NodeResizePending: State set when resize controller has finished resizing the volume but further action is needed on the node. - NodeResizeInProgress: State set when kubelet is resizing the volume. - NodeResizeFailed: State set when kubelet has failed in kubelet with a termination error. Transient errors don't set NodeResizeFailed. example: if expanding a PVC for more capacity, the pvc.status.allocatedResourceStatus['storage.kubernetes.io/resizing'] field can be one of the following states: - ControllerResizeInProgress - pvc.status.allocatedResourceStatus['storage.kubernetes.io/resizing'] = "ControllerResizeInProgress" - ControllerResizeFailed - pvc.status.allocatedResourceStatus['storage.kubernetes.io/resizing'] = "ControllerResizeFailed"

Property	Type	Description
		"ControllerResizeFailed" - pvc.status.allocatedResourceStatus["storage"] "NodeResizePending" - pvc.status.allocatedResourceStatus["storage"] "NodeResizeInProgress" - pvc.status.allocatedResourceStatus["storage"] "NodeResizeFailed" When this field is not null means that no resize operation is in progress for the given PVC. A controller that receives PVC update with unknown resourceName or ClaimResourceName should ignore the update for the purpose it was designed. For example - a controller that is responsible for resizing capacity of the volume should ignore PVC updates that change other values associated with PVC. This is an alpha field and requires enabling RecoverVolumeExpansionFailure feature.
allocatedResources	object	allocatedResources tracks the resources allocated to a PVC including its capacity. Key names follow standard Kubernetes label syntax. Valid values are either: * Un-prefixed keys: - storage - the capacity of the volume. * Custom resources must use implementation-defined prefixed names such as "example.com/my-custom-resource" Apart from these values - keys that are unprefixed or have kubernetes.io prefix are considered reserved and hence may not be used. Capacity reported here may be larger than the capacity when a volume expansion operation is requested. For storage quota, the larger value

Property	Type	Description
		allocatedResources and PVC.spec.resour If allocatedResources is not set, PVC.spec alone is used for quota calculation. If a vol expansion capacity request is lowered, allocatedResources is only lowered if ther expansion operations in progress and if th volume capacity is equal or lower than the capacity. A controller that receives PVC update with unknown resourceName should ignore the the purpose it was designed. For example controller that only is responsible for resizi of the volume, should ignore PVC updates change other valid resources associated v This is an alpha field and requires enablin RecoverVolumeExpansionFailure feature.
capacity	object	capacity represents the actual resources c underlying volume.
conditions	array	conditions is the current Condition of persi volume claim. If underlying persistent volu resized then the Condition will be set to 'R

Property	Type	Description
<code>currentVolumeAttributesClassName</code>	<code>string</code>	<code>currentVolumeAttributesClassName</code> is the name of the <code>VolumeAttributesClass</code> the <code>PV</code> When unset, there is no <code>VolumeAttributeC</code> to this <code>PersistentVolumeClaim</code> This is a be requires enabling <code>VolumeAttributesClass</code> f by default).
<code>modifyVolumeStatus</code>	<code>object</code>	<code>ModifyVolumeStatus</code> represents the status <code>ControllerModifyVolume</code> operation. When there is no <code>ModifyVolume</code> operation being This is a beta field and requires enabling <code>VolumeAttributesClass</code> feature (off by defe
<code>phase</code>	<code>string</code>	<code>phase</code> represents the current phase of <code>PersistentVolumeClaim</code> .

`.spec.mysql.mgr.backupVolumeClaimTemplate.status.accessModes`

Description

`accessModes` contains the actual access modes the volume backing the PVC has. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1>

Type

`array`

`.spec.mysql.mgr.backupVolumeClaimTemplate.status.accessModes[]`

Type

string

.spec.mysql.mgr.backupVolumeClaimTemplate.status.allocatedResourceStatuses**Description**

allocatedResourceStatuses stores status of resource being resized for the given PVC. Key names follow standard Kubernetes label syntax. Valid values are either: * Un-prefixed keys: - storage - the capacity of the volume. * Custom resources must use implementation-defined prefixed names such as "example.com/my-custom-resource" Apart from above values - keys that are unprefixed or have kubernetes.io prefix are considered reserved and hence may not be used. ClaimResourceStatus can be in any of following states: -

ControllerResizeInProgress: State set when resize controller starts resizing the volume in control-plane. - ControllerResizeFailed: State set when resize has failed in resize controller with a terminal error. - NodeResizePending: State set when resize controller has finished resizing the volume but further resizing of volume is needed on the node. -

NodeResizeInProgress: State set when kubelet starts resizing the volume. -

NodeResizeFailed: State set when resizing has failed in kubelet with a terminal error.

Transient errors don't set NodeResizeFailed. For example: if expanding a PVC for more capacity - this field can be one of the following states: -

pvc.status.allocatedResourceStatus[storage] = "ControllerResizeInProgress" -

pvc.status.allocatedResourceStatus[storage] = "ControllerResizeFailed" -

pvc.status.allocatedResourceStatus[storage] = "NodeResizePending" -

pvc.status.allocatedResourceStatus[storage] = "NodeResizeInProgress" -

pvc.status.allocatedResourceStatus[storage] = "NodeResizeFailed" When this field is not set, it means that no resize operation is in progress for the given PVC. A controller that receives PVC update with previously unknown resourceName or ClaimResourceStatus should ignore the update for the purpose it was designed. For example - a controller that only is responsible for resizing capacity of the volume, should ignore PVC updates that change other valid resources associated with PVC. This is an alpha field and requires enabling RecoverVolumeExpansionFailure feature.

Type

object

.spec.mysql.mgr.backupVolumeClaimTemplate.status.allocatedResources

Description

allocatedResources tracks the resources allocated to a PVC including its capacity. Key names follow standard Kubernetes label syntax. Valid values are either: * Un-prefixed keys: - storage - the capacity of the volume. * Custom resources must use implementation-defined prefixed names such as "example.com/my-custom-resource" Apart from above values - keys that are unprefixed or have kubernetes.io prefix are considered reserved and hence may not be used. Capacity reported here may be larger than the actual capacity when a volume expansion operation is requested. For storage quota, the larger value from allocatedResources and PVC.spec.resources is used. If allocatedResources is not set, PVC.spec.resources alone is used for quota calculation. If a volume expansion capacity request is lowered, allocatedResources is only lowered if there are no expansion operations in progress and if the actual volume capacity is equal or lower than the requested capacity. A controller that receives PVC update with previously unknown resourceName should ignore the update for the purpose it was designed. For example - a controller that only is responsible for resizing capacity of the volume, should ignore PVC updates that change other valid resources associated with PVC. This is an alpha field and requires enabling RecoverVolumeExpansionFailure feature.

Type

object

.spec.mysql.mgr.backupVolumeClaimTemplate.status.capacity

Description

capacity represents the actual resources of the underlying volume.

Type

object

.spec.mysql.mgr.backupVolumeClaimTemplate.status.conditions

Description

conditions is the current Condition of persistent volume claim. If underlying persistent volume is being resized then the Condition will be set to 'Resizing'.

Type

array

.spec.mysql.mgr.backupVolumeClaimTemplate.status.conditions[]

Description

PersistentVolumeClaimCondition contains details about state of pvc

Type

object

Required

status type

Property	Type	Description
lastProbeTime	string	lastProbeTime is the time we probed the condition.
lastTransitionTime	string	lastTransitionTime is the time the condition transitioned fro
message	string	message is the human-readable message indicating detail
reason	string	reason is a unique, this should be a short, machine unders the reason for condition's last transition. If it reports "Resizi underlying persistent volume is being resized.

Property	Type	Description
status	string	Status is the status of the condition. Can be True, False, U https://kubernetes.io/docs/reference/kubernetes-api/config-resources/persistent-volume-claim-v1/#:~:text=state%20of%20pvc-,conditions.status,-(string)%
type	string	Type is the type of the condition. More info: https://kubernetes.io/docs/reference/kubernetes-api/config-resources/persistent-volume-claim-v1/#:~:text=set%20to%20%27ResizeStarted%27.-,Persistecontains%20details%20about

```
.spec.mysql.mgr.backupVolumeClaimTemplate.status.modifiedVolumeStatus
```

Description

ModifyVolumeStatus represents the status object of ControllerModifyVolume operation. When this is unset, there is no ModifyVolume operation being attempted. This is a beta field and requires enabling VolumeAttributesClass feature (off by default).

Type

object

Required

status

Property	Type	Description
status	string	status is the status of the ControllerModifyVolume operation. It can be in any of following states:

Property	Type	Description
		Pending Pending indicates that the PersistentVolumeClaim cannot be modified due to unmet requirements, such as the specified VolumeAttributesClass not existing. InProgress InProgress indicates that the volume is being modified. Infeasible Infeasible indicates that the request has been rejected as invalid by the CSI driver. To resolve the error, a valid VolumeAttributesClass needs to be specified. Note: New statuses can be added in the future. Consumers should check for unknown statuses and fail appropriately.
<code>targetVolumeAttributesClassName</code>	<code>string</code>	targetVolumeAttributesClassName is the name of the VolumeAttributesClass the PVC currently being reconciled

.spec.mysql.mgr.certConfig

Description

CertConfig allows a user to specify custom CA certificate with advanced option.

Type

`object`

Property	Type	Description
<code>DNSNames</code>	<code>array</code>	DNSNames means the dns name list of the cert.
<code>duration</code>	<code>string</code>	Duration means the duration of the cert.
<code>secretName</code>	<code>string</code>	SecretName means the secret of the cert.

`.spec.mysql.mgr.certConfig.DNSNames`

Description

DNSNames means the dns name list of the cert.

Type

`array`

`.spec.mysql.mgr.certConfig.DNSNames[]`

Type

`string`

`.spec.mysql.mgr.config`

Description

Config allows a user to specify a custom configuration file for MySQL.

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

`.spec.mysql.mgr.image`

Description

Image allows a user to specify image.

Type

`object`

Property	Type	Description
<code>agentImage</code>	<code>string</code>	AgentImage means the image of the agent.
<code>routerImage</code>	<code>string</code>	RouterImage means the image of the router.
<code>serverImage</code>	<code>string</code>	ServerImage means the image of the server.
<code>slowSQL</code>	<code>string</code>	SlowSQL means the image of the slow sql.

`.spec.mysql.mgr.imagePullSecret`

Description

ImagePullSecret defines the name of the secret that contains the required credentials for pulling from the specified Repository.

Type

array

.spec.mysql.mgr.imagePullSecret[]

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

.spec.mysql.mgr.monitor

Description

Monitor define MySQL monitor spec

Type

object

Property	Type	Description
<code>enable</code>	<code>boolean</code>	Enable means enable monitor
<code>exporter</code>	<code>object</code>	Exporter define MySQL exporter spec

`.spec.mysql.mgr.monitor.exporter`

Description

Exporter define MySQL exporter spec

Type

`object`

Property	Type	Description
<code>image</code>	<code>string</code>	Image means the image of the exporter.
<code>resources</code>	<code>object</code>	Resources holds ResourceRequirements for the MySQL Agent & Server Containers.

`.spec.mysql.mgr.monitor.exporter.resources`

Description

Resources holds ResourceRequirements for the MySQL Agent & Server Containers.

Type

`object`

Property	Type	Description
		Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container.
claims	array	This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
limits	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗
requests	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

.spec.mysql.mgr.monitor.exporter.resources.claims

Description

Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.

Type

array

.spec.mysql.mgr.monitor.exporter.resources.claims[]

Description

ResourceClaim references one entry in PodSpec.ResourceClaims.

Type

object

Required

name

Property	Type	Description
name	string	Name must match the name of one entry in pod.spec.resourceClaims of the Pod where this field is used. It makes that resource available inside a container.
request	string	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

.spec.mysql.mgr.monitor.exporter.resources.limits

Description

Limits describes the maximum amount of compute resources allowed. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

.spec.mysql.mgr.monitor.exporter.resources.requests

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info:

<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

.spec.mysql.mgr.multiClusterReplication

Description

Config this field to support multi cluster replication for disaster recovery.

Type

object

Property	Type	Description
<code>bootstrapSecret</code>	<code>string</code>	BootstrapSecret means the secret of the multi cluster replication.
<code>clusterSetSeedAddr</code>	<code>string</code>	ClusterSetSeedAddr means the cluster set seed address of the multi cluster replication.
<code>enabled</code>	<code>boolean</code>	Enabled means enable multi cluster replication
<code>hostPort</code>	<code>integer</code>	HostPort means the host port of the multi cluster replication.

.spec.mysql.mgr.nodeSelector

Description

NodeSelector is a selector which must be true for the pod to fit on a node. Selector which must match a node's labels for the pod to be scheduled on that node. More info: <https://kubernetes.io/docs/concepts/configuration/assign-pod-node/>

Type

object

.spec.mysql.mgr.paras

Description

Paras provide paras template feature

Type

object

.spec.mysql.mgr.patches

Description

Patches use to apply patches to the MySQL container

Type

object

Property	Type	Description
podPatches	object	PodPatches is a map of pod name to patch
servicePatches	object	ServicePatches is a map of service name to patch

.spec.mysql.mgr.patches.podPatches

Description

PodPatches is a map of pod name to patch

Type

object

.spec.mysql.mgr.patches.servicePatches

Description

ServicePatches is a map of service name to patch

Type

object

.spec.mysql.mgr.resources

Description

Resources holds ResourceRequirements for the MySQL Agent & Server Containers

Type

object

Property	Type	Description
agent	object	Agent describes the compute resource requirements for agent.
server	object	Server describes the compute resource requirements for server.

.spec.mysql.mgr.resources.agent

Description

Agent describes the compute resource requirements for agent.

Type

object

Property	Type	Description
		Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container.
claims	array	This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
limits	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗
requests	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

.spec.mysql.mgr.resources.agent.claims

Description

Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.

Type

array

.spec.mysql.mgr.resources.agent.claims[]

Description

ResourceClaim references one entry in PodSpec.ResourceClaims.

Type

object

Required

name

Property	Type	Description
name	string	Name must match the name of one entry in pod.spec.resourceClaims of the Pod where this field is used. It makes that resource available inside a container.
request	string	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

.spec.mysql.mgr.resources.agent.limits

Description

Limits describes the maximum amount of compute resources allowed. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

.spec.mysql.mgr.resources.agent.requests

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info:

<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

.spec.mysql.mgr.resources.server

Description

Server describes the compute resource requirements for server.

Type

object

Property	Type	Description
		Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container.
claims	array	This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
limits	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗
requests	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to

Property	Type	Description
		Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

.spec.mysql.mgr.resources.server.claims

Description

Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.

Type

array

.spec.mysql.mgr.resources.server.claims[]

Description

ResourceClaim references one entry in PodSpec.ResourceClaims.

Type

object

Required

name

Property	Type	Description
name	string	Name must match the name of one entry in pod.spec.resourceClaims of the Pod where this field is used. It makes that resource available inside a container.

Property	Type	Description
<code>request</code>	<code>string</code>	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

`.spec.mysql.mgr.resources.server.limits`

Description

Limits describes the maximum amount of compute resources allowed. More info:
<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.mysql.mgr.resources.server.requests`

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info:
<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.mysql.mgr.rootPasswordSecret`

Description

If defined, we use this secret for configuring the MYSQL_ROOT_PASSWORD If it is not set we generate a secret dynamically

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

.spec.mysql.mgr.router

Description

Router defines the MySQL Router

Type

`object`

Property	Type	Description
<code>affinity</code>	<code>object</code>	If specified, affinity will define the pod's scheduling constraints
<code>configuration</code>	<code>string</code>	Configuration means the configuration of the router.
<code>mysqlPasswordSecret</code>	<code>string</code>	MySQLPasswordSecret defines the secret name of MySQL user.
<code>mysqlUser</code>	<code>string</code>	MySQLUser defines the user that connect to MySQL.

Property	Type	Description
<code>nodeSelector</code>	<code>object</code>	NodeSelector is a selector which must be true for the pod to fit on a node. Selector which must match a node's labels for the pod to be scheduled on that node. More info: https://kubernetes.io/docs/concepts/configuration/assign-pod-node/ ↗
<code>replicas</code>	<code>integer</code>	Replicas defines the replicas of the MySQL Router.
<code>resources</code>	<code>object</code>	Resources holds ResourceRequirements for the MySQL Agent & Server Containers
<code>securityContext</code>	<code>object</code>	PodSecurityContext holds pod-level security attributes and common container settings. Some fields are also present in container.securityContext. Field values of container.securityContext take precedence over field values of PodSecurityContext.
<code>svcRO</code>	<code>object</code>	SvcRo defines the read only service of the MySQL Router.
<code>svcRW</code>	<code>object</code>	SvcRW defines the read write service of the MySQL Router.
<code>tolerations</code>	<code>array</code>	Tolerations allows specifying a list of tolerations for controlling which set of Nodes a Pod can be scheduled

Property	Type	Description
		on If specified, configuration will define the router's configuration.

`.spec.mysql.mgr.router.affinity`

Description

If specified, affinity will define the pod's scheduling constraints

Type

object

Property	Type	Description
<code>nodeAffinity</code>	object	Describes node affinity scheduling rules for the pod.
<code>podAffinity</code>	object	Describes pod affinity scheduling rules (e.g. co-locate this pod in the same node, zone, etc. as some other pod(s)).
<code>podAntiAffinity</code>	object	Describes pod anti-affinity scheduling rules (e.g. avoid putting this pod in the same node, zone, etc. as some other pod(s)).

`.spec.mysql.mgr.router.affinity.nodeAffinity`

Description

Describes node affinity scheduling rules for the pod.

Type

object

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution affinity expressions, etc.) compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest sum are the most preferred.
requiredDuringSchedulingIgnoredDuringExecution	object	If the affinity requirements specified by this field are not met at scheduling

Property	Type	Description
		time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to an update), the system may or may not try to eventually evict the pod from its node.

`.spec.mysql.mgr.router.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest sum are the most preferred.

Type

array

`.spec.mysql.mgr.router.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

An empty preferred scheduling term matches all objects with implicit weight 0 (i.e. it's a no-op). A null preferred scheduling term matches no objects (i.e. is also a no-op).

Type

object

Required

preference

weight

Property	Type	Description
preference	object	A node selector term, associated with the corresponding weight.
weight	integer	Weight associated with matching the corresponding nodeSelectorTerm, in the range 1-100.

.spec.mysql.mgr.router.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference

Description

A node selector term, associated with the corresponding weight.

Type

object

Property	Type	Description
matchExpressions	array	A list of node selector requirements by node's labels.
matchFields	array	A list of node selector requirements by node's fields.

.spec.mysql.mgr.router.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions

Description

A list of node selector requirements by node's labels.

Type

array

.spec.mysql.mgr.router.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[]

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.

Property	Type	Description
<code>values</code>	<code>array</code>	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

`.spec.mysql.mgr.router.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values`

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.mysql.mgr.router.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values[]`

Type

`string`

`.spec.mysql.mgr.router.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchF`

fields

Description

A list of node selector requirements by node's fields.

Type

array

.spec.mysql.mgr.router.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[]

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will

Property	Type	Description
		be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.mysql.mgr.router.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.mgr.router.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values[]

Type

string

.spec.mysql.mgr.router.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to

be met at some point during pod execution (e.g. due to an update), the system may or may not try to eventually evict the pod from its node.

Type

object

Required

nodeSelectorTerms

Property	Type	Description
nodeSelectorTerms	array	Required. A list of node selector terms. The terms are ORed.

.spec.mysql.mgr.router.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms

Description

Required. A list of node selector terms. The terms are ORed.

Type

array

.spec.mysql.mgr.router.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[]

Description

A null or empty node selector term matches no objects. The requirements of them are ANDed. The TopologySelectorTerm type implements a subset of the NodeSelectorTerm.

Type

object

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	A list of node selector requirements by node's labels.
<code>matchFields</code>	<code>array</code>	A list of node selector requirements by node's fields.

`.spec.mysql.mgr.router.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions`

Description

A list of node selector requirements by node's labels.

Type

`array`

`.spec.mysql.mgr.router.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[]`

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

`key`

`operator`

Property	Type	Description
<code>key</code>	<code>string</code>	The label key that the selector applies to.
<code>operator</code>	<code>string</code>	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
<code>values</code>	<code>array</code>	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

`.spec.mysql.mgr.router.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values`

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.mysql.mgr.router.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[`

`].matchExpressions[].values[]`

Type

`string`

`.spec.mysql.mgr.router.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields`

Description

A list of node selector requirements by node's fields.

Type

`array`

`.spec.mysql.mgr.router.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[]`

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

`key``operator`

Property	Type	Description
<code>key</code>	<code>string</code>	The label key that the selector applies to.

Property	Type	Description
<code>operator</code>	<code>string</code>	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
<code>values</code>	<code>array</code>	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

`.spec.mysql.mgr.router.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values`

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.mysql.mgr.router.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values[]`

Type

`string`

.spec.mysql.mgr.router.affinity.podAffinity

Description

Describes pod affinity scheduling rules (e.g. co-locate this pod in the same node, zone, etc. as some other pod(s)).

Type

object

Property	Type	Description
<code>preferredDuringSchedulingIgnoredDuringExecution</code>	array	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution, affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the

Property	Type	Description
		node(s) with the highest sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	If the affinity requirement: specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirement: specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

`.spec.mysql.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights,

i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

array

.spec.mysql.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

Description

The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Type

object

Required

podAffinityTerm weight

Property	Type	Description
podAffinityTerm	object	Required. A pod affinity term, associated with the corresponding weight.
weight	integer	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

.spec.mysql.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm

Description

Required. A pod affinity term, associated with the corresponding weight.

Type

object

Required

topologyKey

Property	Type	Description
labelSelector	object	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
matchLabelKeys	array	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
mismatchLabelKeys	array	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with

Property	Type	Description
		<code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both <code>mismatchLabelKeys</code> and <code>labelSelector</code> . Also, <code>mismatchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. This is a beta field and requires enabling <code>MatchLabelKeysInPodAffinity</code> feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the <code>namespaces</code> field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector (<code>{}</code>) matches all namespaces.
<code>namespaces</code>	<code>array</code>	<code>namespaces</code> specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by <code>namespaceSelector</code> . null or empty namespaces list and null <code>namespaceSelector</code> means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the <code>labelSelector</code> in the specified namespaces, where co-located is defined as running on a node whose value of the label with key <code>topologyKey</code> matches that of any

Property	Type	Description
		node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.mysql.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.mysql.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mysql.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]

Type

string

.spec.mysql.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

`.spec.mysql.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys`

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

`.spec.mysql.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys`

Type

string

`.spec.mysql.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys`

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti)

affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both `mismatchLabelKeys` and `labelSelector`. Also, `mismatchLabelKeys` cannot be set when `labelSelector` isn't set. This is a beta field and requires enabling `MatchLabelKeysInPodAffinity` feature gate (enabled by default).

Type

array

`.spec.mysql.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]`

Type

string

`.spec.mysql.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the `namespaces` field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector (`{}`) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.

Property	Type	Description
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces.selector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.mysql.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces.selector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

`key`

`operator`

Property	Type	Description
<code>key</code>	<code>string</code>	key is the label key that the selector applies to.
<code>operator</code>	<code>string</code>	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
<code>values</code>	<code>array</code>	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.mysql.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces.selector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.mysql.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces.selector.matchExpressions[].values[]`

Type

string

.spec.mysql.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mysql.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.mysql.mgr.router.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

string

`.spec.mysql.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution`

Description

If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Type

array

`.spec.mysql.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[]`

Description

Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running

Type

object

Required

topologyKey

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Property	Type	Description
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Property	Type	Description
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector (<code>{}</code>) matches all namespaces.
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.mysql.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.mysql.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mysql.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.mysql.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]`

Type

string

`.spec.mysql.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

`.spec.mysql.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys`

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

`.spec.mysql.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]`

Type

string

`.spec.mysql.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys`

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

`.spec.mysql.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]`

Type

string

`.spec.mysql.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.mysql.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.mysql.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mysql.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]

Type

string

.spec.mysql.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mysql.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

`.spec.mysql.mgr.router.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]`

Type

`string`

`.spec.mysql.mgr.router.affinity.podAntiAffinity`

Description

Describes pod anti-affinity scheduling rules (e.g. avoid putting this pod in the same node, zone, etc. as some other pod(s)).

Type

`object`

Property	Type	Description
<code>preferredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, <code>requiredDuringSchedulingIgnoredDuringExecution</code> anti-affinity expressions, etc.), compute a sum by

Property	Type	Description
		iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

`.spec.mysql.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, `requiredDuringScheduling` anti-affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding `podAffinityTerm`; the node(s) with the highest sum are the most preferred.

Type

`array`

`.spec.mysql.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

The weights of all of the matched `WeightedPodAffinityTerm` fields are added per-node to find the most preferred node(s)

Type

`object`

Required

`podAffinityTerm``weight`

Property	Type	Description
<code>podAffinityTerm</code>	<code>object</code>	Required. A pod affinity term, associated with the corresponding weight.

Property	Type	Description
<code>weight</code>	<code>integer</code>	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

`.spec.mysql.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm`

Description

Required. A pod affinity term, associated with the corresponding weight.

Type

`object`

Required

`topologyKey`

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be

Property	Type	Description
		ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Property	Type	Description
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.mysql.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.mysql.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mysql.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm

m.labelSelector.matchExpressions[].values[]

Type

string

.spec.mysql.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mysql.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

`.spec.mysql.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]`

Type

string

`.spec.mysql.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys`

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

`.spec.mysql.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]`

Type

string

`.spec.mysql.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.mysql.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mysql.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm

m.namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]

Type

string

.spec.mysql.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mysql.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.mysql.mgr.router.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

string

.spec.mysql.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Type

array

.spec.mysql.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

Defines a set of pods (namely those matching the `labelSelector` relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key `<topologyKey>` matches that of any node on which a pod of the set of pods is running

Type

object

Required

topologyKey

Property	Type	Description
<code>labelSelector</code>	object	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	array	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>mismatchLabelKeys</code>	array	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The

Property	Type	Description
		keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both <code>mismatchLabelKeys</code> and <code>labelSelector</code>. Also, <code>mismatchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. This is a beta field and requires enabling <code>MatchLabelKeysInPodAffinity</code> feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the <code>namespaces</code> field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector (<code>{}</code>) matches all namespaces.
<code>namespaces</code>	<code>array</code>	<code>namespaces</code> specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by <code>namespaceSelector</code>. null or empty namespaces list and null <code>namespaceSelector</code> means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the <code>labelSelector</code> in the specified namespaces, where co-

Property	Type	Description
		located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.mysql.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.mysql.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mysql.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]

Type

string

.spec.mysql.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

`.spec.mysql.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys`

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

`.spec.mysql.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]`

Type

string

`.spec.mysql.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys`

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti)

affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both `mismatchLabelKeys` and `labelSelector`. Also, `mismatchLabelKeys` cannot be set when `labelSelector` isn't set. This is a beta field and requires enabling `MatchLabelKeysInPodAffinity` feature gate (enabled by default).

Type

array

`.spec.mysql.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]`

Type

string

`.spec.mysql.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the `namespaces` field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector (`{}`) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.

Property	Type	Description
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.mysql.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

`key`

`operator`

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mysql.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]

Type

`string`

.spec.mysql.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

`object`

.spec.mysql.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

`array`

.spec.mysql.mgr.router.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]

Type

`string`

.spec.mysql.mgr.router.nodeSelector

Description

NodeSelector is a selector which must be true for the pod to fit on a node. Selector which must match a node's labels for the pod to be scheduled on that node. More info: <https://kubernetes.io/docs/concepts/configuration/assign-pod-node/>

Type

object

.spec.mysql.mgr.router.resources

Description

Resources holds ResourceRequirements for the MySQL Agent & Server Containers

Type

object

Property	Type	Description
		Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container.
claims	array	This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
limits	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗
requests	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an

Property	Type	Description
		implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

.spec.mysql.mgr.router.resources.claims

Description

Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.

Type

array

.spec.mysql.mgr.router.resources.claims[]

Description

ResourceClaim references one entry in PodSpec.ResourceClaims.

Type

object

Required

name

Property	Type	Description
name	string	Name must match the name of one entry in pod.spec.resourceClaims of the Pod where this field is used. It makes that resource available inside a container.

Property	Type	Description
<code>request</code>	<code>string</code>	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

`.spec.mysql.mgr.router.resources.limits`

Description

Limits describes the maximum amount of compute resources allowed. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.mysql.mgr.router.resources.requests`

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.mysql.mgr.router.securityContext`

Description

PodSecurityContext holds pod-level security attributes and common container settings. Some fields are also present in container.securityContext. Field values of container.securityContext take precedence over field values of PodSecurityContext.

Type

object

Property	Type	Description
appArmorProfile	object	appArmorProfile is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.
fsGroup	integer	A special supplemental group that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod: 1. The owning GID will be the FSGroup 2. The setgid bit is set (new files created in the volume will be owned by FSGroup) 3. The permission bits are OR'd with rw-rw-- -- If unset, the Kubelet will not modify the ownership and permissions of any volume. Note that this field cannot be set when spec.os.name is windows.
fsGroupChangePolicy	string	fsGroupChangePolicy defines behavior of changing ownership and permission of the volume before being exposed inside Pod. This field will only apply to volume types which support fsGroup based ownership(and permissions). It will have no effect on ephemeral volume types such as: secret, configmaps and emptydir. Valid values are "OnRootMismatch" and "Always". If not

Property	Type	Description
		specified, "Always" is used. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this

Property	Type	Description
		field cannot be set when spec.os.name is windows.
seLinuxChangePolicy	string	seLinuxChangePolicy defines how the container's SELinux label is applied to all volumes used by the Pod. It has no effect on nodes that do not support SELinux or to volumes does not support SELinux. Valid values are "MountOption" and "Recursive". "Recursive" means relabeling of all files on all Pod volumes by the container runtime. This may be slow for large volumes, but allows mixing privileged and unprivileged Pods sharing the same volume on the same node. "MountOption" mounts all eligible Pod volumes with <code>-o context</code> mount option. This requires all Pods that share the same volume to use the same SELinux label. It is not possible to share the same volume among privileged and unprivileged Pods. Eligible volumes are in-tree FibreChannel and iSCSI volumes, and all CSI volumes whose CSI driver announces SELinux support by setting spec.seLinuxMount: true in their CSIDriver instance. Other volumes are always re-labelled recursively. "MountOption" value is allowed only when SELinuxMount feature gate is enabled. If not specified and SELinuxMount feature gate is enabled, "MountOption" is used. If not specified and SELinuxMount feature

Property	Type	Description
		gate is disabled, "MountOption" is used for ReadWriteOncePod volumes and "Recursive" for all other volumes. This field affects only Pods that have SELinux label set, either in PodSecurityContext or in SecurityContext of all containers. All Pods that use the same volume should use the same seLinuxChangePolicy, otherwise some pods can get stuck in ContainerCreating state. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
seccompProfile	object	The seccomp options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
<code>supplementalGroups</code>	<code>array</code>	A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.
<code>supplementalGroupsPolicy</code>	<code>string</code>	Defines how supplemental groups of the first container processes are calculated. Valid values are "Merge" and "Strict". If not specified, "Merge" is used. (Alpha) Using the field requires the SupplementalGroupsPolicy feature gate to be enabled and the container runtime must implement support for this feature. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
<code>sysctls</code>	<code>array</code>	Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.
<code>windowsOptions</code>	<code>object</code>	The Windows specific settings applied to all containers. If unspecified, the options within a container's SecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

`.spec.mysql.mgr.router.securityContext.appArmorProfile`

Description

appArmorProfile is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured

Property	Type	Description
		on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
<code>type</code>	<code>string</code>	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

`.spec.mysql.mgr.router.securityContext.seLinuxOptions`

Description

The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when `spec.os.name` is windows.

Type

`object`

Property	Type	Description
<code>level</code>	<code>string</code>	Level is SELinux level label that applies to the container.
<code>role</code>	<code>string</code>	Role is a SELinux role label that applies to the container.
<code>type</code>	<code>string</code>	Type is a SELinux type label that applies to the container.

Property	Type	Description
<code>user</code>	<code>string</code>	User is a SELinux user label that applies to the container.

`.spec.mysql.mgr.router.securityContext.seccompProfile`

Description

The seccomp options to use by the containers in this pod. Note that this field cannot be set when `spec.os.name` is windows.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.
<code>type</code>	<code>string</code>	<code>type</code> indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied.

.spec.mysql.mgr.router.securityContext.supplementalGroups

Description

A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.

Type

array

.spec.mysql.mgr.router.securityContext.supplementalGroups[]

Type

integer

.spec.mysql.mgr.router.securityContext.sysctls

Description

Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.

Type

array

.spec.mysql.mgr.router.securityContext.sysctls[]

Description

Sysctl defines a kernel parameter to be set

Type

object

Required

name

value

Property	Type	Description
name	string	Name of a property to set
value	string	Value of a property to set

.spec.mysql.mgr.router.securityContext.windowsOptions**Description**

The Windows specific settings applied to all containers. If unspecified, the options within a container's SecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
gmsaCredentialSpec	string	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the GMSACredentialSpecName field.

Property	Type	Description
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

`.spec.mysql.mgr.router.svcRO`

Description

SvcRo defines the read only service of the MySQL Router.

Type

`object`

Property	Type	Description
<code>port</code>	<code>integer</code>	Port means the port of the node port.
<code>type</code>	<code>string</code>	Type means the type of the service.

`.spec.mysql.mgr.router.svcRW`

Description

SvcRW defines the read write service of the MySQL Router.

Type

`object`

Property	Type	Description
<code>port</code>	<code>integer</code>	Port means the port of the node port.
<code>type</code>	<code>string</code>	Type means the type of the service.

`.spec.mysql.mgr.router.tolerations`

Description

Tolerations allows specifying a list of tolerations for controlling which set of Nodes a Pod can be scheduled on. If specified, configuration will define the router's configuration.

Type

`array`

.spec.mysql.mgr.router.tolerations[]

Description

The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.

Type

object

Property	Type	Description
effect	string	Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.
key	string	Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is empty, operator must be Exists; this combination means to match all values and all keys.
operator	string	Operator represents a key's relationship to the value. Valid operators are Exists and Equal. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category.
tolerationSeconds	integer	TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is not set, which means tolerate the taint

Property	Type	Description
		forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.
<code>value</code>	<code>string</code>	Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

`.spec.mysql.mgr.securityContext`

Description

SecurityContext holds Pod-level security attributes and common Container settings.

Type

`object`

Property	Type	Description
<code>appArmorProfile</code>	<code>object</code>	appArmorProfile is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.
<code>fsGroup</code>	<code>integer</code>	A special supplemental group that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod: 1. The owning GID will be the FSGroup 2. The setgid bit is set (new files created in the volume will be owned by FSGroup)

Property	Type	Description
		3. The permission bits are OR'd with rw-rw-- -- If unset, the Kubelet will not modify the ownership and permissions of any volume. Note that this field cannot be set when spec.os.name is windows.
fsGroupChangePolicy	string	fsGroupChangePolicy defines behavior of changing ownership and permission of the volume before being exposed inside Pod. This field will only apply to volume types which support fsGroup based ownership(and permissions). It will have no effect on ephemeral volume types such as: secret, configmaps and emptydir. Valid values are "OnRootMismatch" and "Always". If not specified, "Always" is used. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
<code>runAsNonRoot</code>	<code>boolean</code>	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
<code>runAsUser</code>	<code>integer</code>	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
<code>seLinuxChangePolicy</code>	<code>string</code>	seLinuxChangePolicy defines how the container's SELinux label is applied to all volumes used by the Pod. It has no effect on nodes that do not support SELinux or to volumes does not support SELinux. Valid values are "MountOption" and "Recursive". "Recursive" means relabeling of all files on all Pod volumes by the container runtime. This may be slow for large volumes, but allows mixing privileged and unprivileged

Property	Type	Description
		Pods sharing the same volume on the same node. "MountOption" mounts all eligible Pod volumes with <code>-o context</code> mount option. This requires all Pods that share the same volume to use the same SELinux label. It is not possible to share the same volume among privileged and unprivileged Pods. Eligible volumes are in-tree FibreChannel and iSCSI volumes, and all CSI volumes whose CSI driver announces SELinux support by setting <code>spec.seLinuxMount: true</code> in their CSIDriver instance. Other volumes are always re-labelled recursively. "MountOption" value is allowed only when SELinuxMount feature gate is enabled. If not specified and SELinuxMount feature gate is enabled, "MountOption" is used. If not specified and SELinuxMount feature gate is disabled, "MountOption" is used for <code>ReadWriteOncePod</code> volumes and "Recursive" for all other volumes. This field affects only Pods that have SELinux label set, either in <code>PodSecurityContext</code> or in <code>SecurityContext</code> of all containers. All Pods that use the same volume should use the same <code>seLinuxChangePolicy</code>, otherwise some pods can get stuck in <code>ContainerCreating</code> state. Note that this field cannot be set when <code>spec.os.name</code> is <code>windows</code>.

Property	Type	Description
seLinuxOptions	object	The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
seccompProfile	object	The seccomp options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
<code>supplementalGroups</code>	<code>array</code>	A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.
<code>supplementalGroupsPolicy</code>	<code>string</code>	Defines how supplemental groups of the first container processes are calculated. Valid values are "Merge" and "Strict". If not specified, "Merge" is used. (Alpha) Using the field requires the SupplementalGroupsPolicy feature gate to be enabled and the container runtime must implement support for this feature. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
<code>sysctls</code>	<code>array</code>	Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.
<code>windowsOptions</code>	<code>object</code>	The Windows specific settings applied to all containers. If unspecified, the options within a container's SecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

`.spec.mysql.mgr.securityContext.appArmorProfile`

Description

appArmorProfile is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured

Property	Type	Description
		on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
<code>type</code>	<code>string</code>	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

.spec.mysql.mgr.securityContext.seLinuxOptions

Description

The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Property	Type	Description
<code>level</code>	<code>string</code>	Level is SELinux level label that applies to the container.
<code>role</code>	<code>string</code>	Role is a SELinux role label that applies to the container.
<code>type</code>	<code>string</code>	Type is a SELinux type label that applies to the container.

Property	Type	Description
<code>user</code>	<code>string</code>	User is a SELinux user label that applies to the container.

`.spec.mysql.mgr.securityContext.seccompProfile`

Description

The seccomp options to use by the containers in this pod. Note that this field cannot be set when `spec.os.name` is windows.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.
<code>type</code>	<code>string</code>	<code>type</code> indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied.

.spec.mysql.mgr.securityContext.supplementalGroups

Description

A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.

Type

array

.spec.mysql.mgr.securityContext.supplementalGroups[]

Type

integer

.spec.mysql.mgr.securityContext.sysctls

Description

Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.

Type

array

.spec.mysql.mgr.securityContext.sysctls[]

Description

Sysctl defines a kernel parameter to be set

Type

object

Required

name

value

Property	Type	Description
name	string	Name of a property to set
value	string	Value of a property to set

.spec.mysql.mgr.securityContext.windowsOptions

Description

The Windows specific settings applied to all containers. If unspecified, the options within a container's SecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
gmsaCredentialSpec	string	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.

Property	Type	Description
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entryptoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

`.spec.mysql.mgr.slowSQL`

Description

SlowSQL define MySQL slow sql spec

Type

`object`

Property	Type	Description
<code>enabled</code>	<code>boolean</code>	Enabled means enable slow sql
<code>image</code>	<code>string</code>	Image means the image of the slow sql.
<code>resources</code>	<code>object</code>	Resources holds ResourceRequirements for the MySQL Agent & Server Containers.
<code>serverEndpoint</code>	<code>string</code>	ServerEndpoint means the server endpoint of the slow sql.

`.spec.mysql.mgr.slowSQL.resources`

Description

Resources holds ResourceRequirements for the MySQL Agent & Server Containers.

Type

`object`

Property	Type	Description
<code>claims</code>	<code>array</code>	Claims lists the names of resources, defined in <code>spec.resourceClaims</code>, that are used by this container. This is an alpha field and requires enabling the <code>DynamicResourceAllocation</code> feature gate. This field is immutable. It can only be set for containers.

Property	Type	Description
<code>limits</code>	<code>object</code>	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗
<code>requests</code>	<code>object</code>	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

`.spec.mysql.mgr.slowSQL.resources.claims`

Description

Claims lists the names of resources, defined in `spec.resourceClaims`, that are used by this container. This is an alpha field and requires enabling the `DynamicResourceAllocation` feature gate. This field is immutable. It can only be set for containers.

Type

`array`

`.spec.mysql.mgr.slowSQL.resources.claims[]`

Description

ResourceClaim references one entry in `PodSpec.ResourceClaims`.

Type

`object`

Required

name

Property	Type	Description
name	string	Name must match the name of one entry in pod.spec.resourceClaims of the Pod where this field is used. It makes that resource available inside a container.
request	string	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

.spec.mysql.mgr.slowSQL.resources.limits

Description

Limits describes the maximum amount of compute resources allowed. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

.spec.mysql.mgr.slowSQL.resources.requests

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

.spec.mysql.mgr.sslSecret

Description

SSLSecret allows a user to specify custom CA certificate, server certificate and server key for group replication SSL.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

.spec.mysql.mgr.tolerations

Description

Tolerations allows specifying a list of tolerations for controlling which set of Nodes a Pod can be scheduled on

Type

array

.spec.mysql.mgr.tolerations[]

Description

The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.

Type

object

Property	Type	Description
<code>effect</code>	<code>string</code>	Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.
<code>key</code>	<code>string</code>	Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is empty, operator must be Exists; this combination means to match all values and all keys.
<code>operator</code>	<code>string</code>	Operator represents a key's relationship to the value. Valid operators are Exists and Equal. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category.
<code>tolerationSeconds</code>	<code>integer</code>	TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is not set, which means tolerate the taint forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.
<code>value</code>	<code>string</code>	Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

.spec.mysql.mgr.upgradeOption

Description

UpgradeOption defines the upgrade option for the MySQL cluster.

Type

object

Property	Type	Description
<code>autoUpgrade</code>	<code>boolean</code>	AutoUpgrade means the auto upgrade option
<code>crVersion</code>	<code>string</code>	CRVersion means the version of the CR

.spec.mysql.mgr.volumeClaimTemplate

Description

VolumeClaimTemplate allows a user to specify how volumes inside a MySQL cluster

Type

object

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗

Property	Type	Description
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
<code>metadata</code>	<code>ObjectMeta</code> ↗	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata ↗
<code>spec</code>	<code>object</code>	spec defines the desired characteristics of a volume requested by a pod author. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims ↗
<code>status</code>	<code>object</code>	status represents the current information/status of a persistent volume claim. Read-only. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims ↗

`.spec.mysql.mgr.volumeClaimTemplate.spec`

Description

spec defines the desired characteristics of a volume requested by a pod author. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims>

Type

object

Property	Type	Description
accessModes	array	accessModes contains the desired access modes volume should have. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1
dataSource	object	dataSource field can be used to specify either: • An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) • An existing PVC (PersistentVolumeClaim) If the provisioner or an external controller can support the specified data source, it will create a new volume based on the contents of the specified data source. When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If the namespace is specified, then dataSourceRef will not be copied to dataSource.
dataSourceRef	object	dataSourceRef specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a PersistentVolumeClaim object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replace the functionality of the dataSource field and as such

Property	Type	Description
		both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in dataSourceRef, dataSource isn't set to the same value and must be empty. There are three important differences between dataSource and dataSourceRef: • While dataSource only allows two specific types of objects, dataSourceRef allows any non-core object as well as PersistentVolumeClaim objects. • While dataSource ignores disallowed values (dropping them), dataSourceRef preserves all values, and generates an error if a disallowed value is specified. • While dataSource only allows local objects, dataSourceRef allows objects in any namespace (Beta) Using this field requires the AnyVolumeDataSource feature gate to be enabled (Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource feature gate to be enabled.
resources	object	resources represents the minimum resources the volume should have. If RecoverVolumeExpansionFailure feature is enabled, users are allowed to specify resource requirement that are lower than previous value but must still be higher than capacity recorded in the status field of claim. More info:

Property	Type	Description
		https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources ↗
<code>selector</code>	<code>object</code>	selector is a label query over volumes to consider binding.
<code>storageClassName</code>	<code>string</code>	storageClassName is the name of the StorageClass required by the claim. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#class-1 ↗

Property	Type	Description
<code>volumeAttributesClassName</code>	<code>string</code>	<code>volumeAttributesClassName</code> may be used to set the <code>VolumeAttributesClass</code> used by this claim. If specified, the CSI driver will create or update the volume with attributes defined in the corresponding <code>VolumeAttributesClass</code>. This has a different purpose than <code>storageClassName</code>, it can be changed after the claim is created. An empty string value means that <code>VolumeAttributesClass</code> will be applied to the claim if it's not allowed to reset this field to empty string once it is set. If unspecified and the <code>PersistentVolumeClaim</code> is unbound, the default <code>VolumeAttributesClass</code> will be used by the <code>persistentvolume</code> controller if it exists. If the resource referred to by <code>volumeAttributesClass</code> does not exist, this <code>PersistentVolumeClaim</code> will be set to <code>Pending</code> state, as reflected by the <code>modifyVolumeStatus</code> field, until such as a resource exists. More info: https://kubernetes.io/docs/concepts/storage/volume-attributes-classes/ (Beta) Using this field requires the <code>VolumeAttributesClass</code> feature gate to be enabled (off by default).
<code>volumeMode</code>	<code>string</code>	<code>volumeMode</code> defines what type of volume is required by the claim. Value of <code>Filesystem</code> is implied when not included in claim spec.
<code>volumeName</code>	<code>string</code>	<code>volumeName</code> is the binding reference to the <code>PersistentVolume</code> backing this claim.

`.spec.mysql.mgr.volumeClaimTemplate.spec.accessMode`

S

Description

accessModes contains the desired access modes the volume should have. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1>

Type

array

`.spec.mysql.mgr.volumeClaimTemplate.spec.accessMode`

s[]

Type

string

`.spec.mysql.mgr.volumeClaimTemplate.spec.dataSource`

Description

dataSource field can be used to specify either: * An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) * An existing PVC (PersistentVolumeClaim) If the provisioner or an external controller can support the specified data source, it will create a new volume based on the contents of the specified data source. When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If the namespace is specified, then dataSourceRef will not be copied to dataSource.

Type

object

Required

kind

name

Property	Type	Description
<code>apiGroup</code>	<code>string</code>	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
<code>kind</code>	<code>string</code>	Kind is the type of resource being referenced
<code>name</code>	<code>string</code>	Name is the name of resource being referenced

`.spec.mysql.mgr.volumeClaimTemplate.spec.dataSourceRef`

Description

`dataSourceRef` specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a `PersistentVolumeClaim` object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replace the functionality of the `dataSource` field and as such if both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in `dataSourceRef`, both fields (`dataSource` and `dataSourceRef`) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in `dataSourceRef`, `dataSource` isn't set to the same value and must be empty. There are three important differences between `dataSource` and `dataSourceRef`:

- * While `dataSource` only allows two specific types of objects, `dataSourceRef` allows any non-core object, as well as `PersistentVolumeClaim` objects.
- * While `dataSource` ignores disallowed values (dropping them), `dataSourceRef` preserves all values, and generates an error if a disallowed value is specified.
- * While `dataSource` only allows local objects, `dataSourceRef` allows objects in any namespaces.

(Beta) Using this field requires the `AnyVolumeDataSource` feature gate to be enabled.
 (Alpha) Using the namespace field of `dataSourceRef` requires the `CrossNamespaceVolumeDataSource` feature gate to be enabled.

Type

object

Required

kind

name

Property	Type	Description
apiGroup	string	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
kind	string	Kind is the type of resource being referenced
name	string	Name is the name of resource being referenced
namespace	string	Namespace is the namespace of resource being referenced Note that when a namespace is specified, a gateway.networking.k8s.io/ReferenceGrant object is required in the referent namespace to allow that namespace's owner to accept the reference. See the ReferenceGrant documentation for details. (Alpha) This field requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

.spec.mysql.mgr.volumeClaimTemplate.spec.resources

Description

resources represents the minimum resources the volume should have. If RecoverVolumeExpansionFailure feature is enabled users are allowed to specify resource

requirements that are lower than previous value but must still be higher than capacity recorded in the status field of the claim. More info:

<https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources>

Type

object

Property	Type	Description
<code>limits</code>	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗
<code>requests</code>	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

`.spec.mysql.mgr.volumeClaimTemplate.spec.resources.limits`

Description

Limits describes the maximum amount of compute resources allowed. More info:
<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

.spec.mysql.mgr.volumeClaimTemplate.spec.resources.requests

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info:

<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

.spec.mysql.mgr.volumeClaimTemplate.spec.selector

Description

selector is a label query over volumes to consider for binding.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.mysql.mgr.volumeClaimTemplate.spec.selector.matchExpressions

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.mysql.mgr.volumeClaimTemplate.spec.selector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.

Property	Type	Description
<code>values</code>	<code>array</code>	<code>values</code> is an array of string values. If the operator is <code>In</code> or <code>NotIn</code> , the values array must be non-empty. If the operator is <code>Exists</code> or <code>DoesNotExist</code> , the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.mysql.mgr.volumeClaimTemplate.spec.selector.matchExpressions[].values`

Description

`values` is an array of string values. If the operator is `In` or `NotIn`, the values array must be non-empty. If the operator is `Exists` or `DoesNotExist`, the values array must be empty. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.mysql.mgr.volumeClaimTemplate.spec.selector.matchExpressions[].values[]`

Type

`string`

`.spec.mysql.mgr.volumeClaimTemplate.spec.selector.matchLabels`

Description

`matchLabels` is a map of {key,value} pairs. A single {key,value} in the `matchLabels` map is equivalent to an element of `matchExpressions`, whose key field is "key", the operator is `In`, and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mysql.mgr.volumeClaimTemplate.status

Description

status represents the current information/status of a persistent volume claim. Read-only.
More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims>

Type

object

Property	Type	Description
<code>accessModes</code>	array	accessModes contains the actual access modes of the volume backing the PVC has. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1
<code>allocatedResourceStatuses</code>	object	allocatedResourceStatuses stores status of resources being resized for the given PVC. Key names must follow standard Kubernetes label syntax. Valid values are either: * Un-prefixed keys: - storage - the resource of the volume. * Custom resources must use implementation-defined prefixed names such as "example.com/my-custom-resource" Apart from un-prefixed keys, keys that are unprefixed or have kubernetes.io prefix are considered reserved and hence may not be used. ClaimResourceStatus can be in any of following states: - ControllerResizeInProgress: State set when resize controller starts resizing the volume plane. - ControllerResizeFailed: State set when resize has failed in resize controller with a termin

Property	Type	Description
		NodeResizePending: State set when resize has finished resizing the volume but further volume is needed on the node. - NodeResizeInProgress: State set when kubelet is resizing the volume. - NodeResizeFailed: when resizing has failed in kubelet with a permanent error. Transient errors don't set NodeResizeFailed. example: if expanding a PVC for more capacity, the pvc.status.allocatedResourceStatus['storageClassName'] field can be one of the following states: - pvc.status.allocatedResourceStatus['storageClassName'] = "ControllerResizeInProgress" - pvc.status.allocatedResourceStatus['storageClassName'] = "ControllerResizeFailed" - pvc.status.allocatedResourceStatus['storageClassName'] = "NodeResizePending" - pvc.status.allocatedResourceStatus['storageClassName'] = "NodeResizeInProgress" - pvc.status.allocatedResourceStatus['storageClassName'] = "NodeResizeFailed" When this field is not set, it means that no resize operation is in progress for the given PVC. A controller that receives PVC update with unknown resourceName or ClaimResourceName should ignore the update for the purpose it was designed. For example - a controller that is responsible for resizing capacity of the volume should ignore PVC updates that change other values associated with PVC. This is an alpha field and requires enabling the RecoverVolumeExpansionFailure feature.

Property	Type	Description
<code>allocatedResources</code>	object	<code>allocatedResources</code> tracks the resources in a PVC including its capacity. Key names follow standard Kubernetes label syntax. Valid values are either: * Un-prefixed keys: - storage - the capacity of the volume. * Custom resources must use implementation-defined prefixed names such as "example.com/my-custom-resource". Annotations with un-prefixed keys or keys that are unprefixed or have the <code>kubernetes.io</code> prefix are considered reserved and hence may not be used. Capacity reported here may be larger than the capacity when a volume expansion operation is requested. For storage quota, the larger value of <code>allocatedResources</code> and <code>PVC.spec.resources.requests.storage</code> is used. If <code>allocatedResources</code> is not set, <code>PVC.spec.resources.requests.storage</code> alone is used for quota calculation. If a volume expansion capacity request is lowered, <code>allocatedResources</code> is only lowered if there are no expansion operations in progress and if the volume capacity is equal or lower than the requested capacity. A controller that receives PVC update with unknown <code>resourceName</code> should ignore the update for the purpose it was designed. For example, a controller that only is responsible for resizing the volume, should ignore PVC updates that change other valid resources associated with the volume. This is an alpha field and requires enabling the <code>RecoverVolumeExpansionFailure</code> feature.

Property	Type	Description
capacity	object	capacity represents the actual resources consumed by the underlying volume.
conditions	array	conditions is the current Condition of persistentVolumeClaim. If underlying persistent volume is resized then the Condition will be set to 'Resizing'.
currentVolumeAttributesClassName	string	currentVolumeAttributesClassName is the name of the VolumeAttributesClass the PersistentVolumeClaim is requesting to be used to create the PersistentVolume. When unset, there is no VolumeAttributesClass requested for this PersistentVolumeClaim. This is a beta feature and requires enabling VolumeAttributesClass feature (off by default).
modifyVolumeStatus	object	ModifyVolumeStatus represents the status of the ControllerModifyVolume operation. When there is no ModifyVolume operation being performed by the controller, this field is omitted. This is a beta field and requires enabling VolumeAttributesClass feature (off by default).
phase	string	phase represents the current phase of PersistentVolumeClaim.

.spec.mysql.mgr.volumeClaimTemplate.status.accessModes

Description

accessModes contains the actual access modes the volume backing the PVC has. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1>

Type

array

.spec.mysql.mgr.volumeClaimTemplate.status.accessModes[]

Type

string

.spec.mysql.mgr.volumeClaimTemplate.status.allocatedResourceStatuses

Description

allocatedResourceStatuses stores status of resource being resized for the given PVC. Key names follow standard Kubernetes label syntax. Valid values are either: * Un-prefixed keys: - storage - the capacity of the volume. * Custom resources must use implementation-defined prefixed names such as "example.com/my-custom-resource" Apart from above values - keys that are unprefixed or have kubernetes.io prefix are considered reserved and hence may not be used. ClaimResourceStatus can be in any of following states: -

ControllerResizeInProgress: State set when resize controller starts resizing the volume in control-plane. - ControllerResizeFailed: State set when resize has failed in resize controller with a terminal error. - NodeResizePending: State set when resize controller has finished resizing the volume but further resizing of volume is needed on the node. -

NodeResizeInProgress: State set when kubelet starts resizing the volume. -

NodeResizeFailed: State set when resizing has failed in kubelet with a terminal error.

Transient errors don't set NodeResizeFailed. For example: if expanding a PVC for more capacity - this field can be one of the following states: -

pvc.status.allocatedResourceStatus['storage'] = "ControllerResizeInProgress" -

pvc.status.allocatedResourceStatus['storage'] = "ControllerResizeFailed" -

pvc.status.allocatedResourceStatus['storage'] = "NodeResizePending" -

pvc.status.allocatedResourceStatus['storage'] = "NodeResizeInProgress" -

pvc.status.allocatedResourceStatus['storage'] = "NodeResizeFailed" When this field is not set, it means that no resize operation is in progress for the given PVC. A controller that

receives PVC update with previously unknown resourceName or ClaimResourceStatus should ignore the update for the purpose it was designed. For example - a controller that only is responsible for resizing capacity of the volume, should ignore PVC updates that change other valid resources associated with PVC. This is an alpha field and requires enabling RecoverVolumeExpansionFailure feature.

Type

object

.spec.mysql.mgr.volumeClaimTemplate.status.allocatedResources

Description

allocatedResources tracks the resources allocated to a PVC including its capacity. Key names follow standard Kubernetes label syntax. Valid values are either: * Un-prefixed keys: - storage - the capacity of the volume. * Custom resources must use implementation-defined prefixed names such as "example.com/my-custom-resource" Apart from above values - keys that are unprefixed or have kubernetes.io prefix are considered reserved and hence may not be used. Capacity reported here may be larger than the actual capacity when a volume expansion operation is requested. For storage quota, the larger value from allocatedResources and PVC.spec.resources is used. If allocatedResources is not set, PVC.spec.resources alone is used for quota calculation. If a volume expansion capacity request is lowered, allocatedResources is only lowered if there are no expansion operations in progress and if the actual volume capacity is equal or lower than the requested capacity. A controller that receives PVC update with previously unknown resourceName should ignore the update for the purpose it was designed. For example - a controller that only is responsible for resizing capacity of the volume, should ignore PVC updates that change other valid resources associated with PVC. This is an alpha field and requires enabling RecoverVolumeExpansionFailure feature.

Type

object

.spec.mysql.mgr.volumeClaimTemplate.status.capacity

Description

capacity represents the actual resources of the underlying volume.

Type

object

.spec.mysql.mgr.volumeClaimTemplate.status.conditions

Description

conditions is the current Condition of persistent volume claim. If underlying persistent volume is being resized then the Condition will be set to 'Resizing'.

Type

array

.spec.mysql.mgr.volumeClaimTemplate.status.conditions[]

Description

PersistentVolumeClaimCondition contains details about state of pvc

Type

object

Required

status

type

Property	Type	Description
lastProbeTime	string	lastProbeTime is the time we probed the condition.
lastTransitionTime	string	lastTransitionTime is the time the condition transitioned fro

Property	Type	Description
<code>message</code>	<code>string</code>	message is the human-readable message indicating detail
<code>reason</code>	<code>string</code>	reason is a unique, this should be a short, machine unders the reason for condition's last transition. If it reports "Resizi underlying persistent volume is being resized.
<code>status</code>	<code>string</code>	Status is the status of the condition. Can be True, False, U https://kubernetes.io/docs/reference/kubernetes-api/config-resources/persistent-volume-claim-v1/#:~:text=state%20of%20pvc-,conditions.status,-(string)%
<code>type</code>	<code>string</code>	Type is the type of the condition. More info: https://kubernetes.io/docs/reference/kubernetes-api/config-resources/persistent-volume-claim-v1/#:~:text=set%20to%20%27ResizeStarted%27.-,Persiste contains%20details%20about

`.spec.mysql.mgr.volumeClaimTemplate.status.modifyVolumeStatus`

Description

ModifyVolumeStatus represents the status object of ControllerModifyVolume operation. When this is unset, there is no ModifyVolume operation being attempted. This is a beta field and requires enabling VolumeAttributesClass feature (off by default).

Type

`object`

Required

status

Property	Type	Description
status	string	status is the status of the ControllerModifyVolume operation. It can be in any of following states: • Pending Pending indicates that the PersistentVolumeClaim cannot be modified due to unmet requirements, such as the specified VolumeAttributesClass not existing. • InProgress InProgress indicates that the volume is being modified. • Infeasible Infeasible indicates that the request has been rejected as invalid by the CSI driver. To resolve the error, a valid VolumeAttributesClass needs to be specified. Note: New statuses can be added in the future. Consumers should check for unknown statuses and fail appropriately.
targetVolumeAttributesClassName	string	targetVolumeAttributesClassName is the name of the VolumeAttributesClass the PVC currently being reconciled

Description

Params is the configuration of MySQL Server

Type

object

Property	Type	Description
mysql	object	
router	object	

.spec.mysql.params.mysql

Type

object

.spec.mysql.params.router

Type

object

.spec.mysql.paras

Description

Paras deprecated: this field is deprecated, please use spec.params instead

Type

object

.spec.mysql.pxc

Description

PXC use to configure PerconaXtraDBCluster

Type

object

Property	Type	Description
allowUnsafeConfigurations	boolean	
backup	object	
clusterReplication	object	
crVersion	string	
dcDesc	object	
enableCRValidationWebhook	boolean	
haproxy	object	
initImage	string	
ipFamilyPrefer	string	IPFamily represents the IP Family (IPv4 or IPv6). This type is used to express the family of an IP expressed by a type (e.g. service.spec.ipFamilies).
logCollectorSecretName	string	
logcollector	object	
multiDcSpec	object	
mysqldExporter	object	
pause	boolean	
platform	string	
proxysql	object	

Property	Type	Description
<code>pxc</code>	<code>object</code>	
<code>pxcLogPath</code>	<code>string</code>	
<code>restrictedPsaEnabled</code>	<code>boolean</code>	
<code>secretsName</code>	<code>string</code>	
<code>serviceMonitor</code>	<code>object</code>	
<code>slowsql</code>	<code>object</code>	
<code>sslInternalSecretName</code>	<code>string</code>	
<code>sslSecretName</code>	<code>string</code>	
<code>tls</code>	<code>object</code>	
<code>updateStrategy</code>	<code>string</code>	StatefulSetUpdateStrategyType is a string enumeration type that enumerates all possible update strategies for the StatefulSet controller.
<code>upgradeOptions</code>	<code>object</code>	
<code>vaultSecretName</code>	<code>string</code>	

`.spec.mysql.pxc.backup`

Type

`object`

Property	Type	Description
<code>annotations</code>	<code>object</code>	

Property	Type	Description
<code>image</code>	<code>string</code>	
<code>imagePullPolicy</code>	<code>string</code>	PullPolicy describes a policy for if/when to pull a container image
<code>imagePullSecrets</code>	<code>array</code>	
<code>pitr</code>	<code>object</code>	
<code>schedule</code>	<code>array</code>	
<code>serviceAccountName</code>	<code>string</code>	
<code>storages</code>	<code>object</code>	

`.spec.mysql.pxc.backup.annotations`

Type

`object`

`.spec.mysql.pxc.backup.imagePullSecrets`

Type

`array`

`.spec.mysql.pxc.backup.imagePullSecrets[]`

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

`.spec.mysql.pxc.backup.pitr`

Type

`object`

Required

`enabled``storageName`

Property	Type	Description
<code>enabled</code>	<code>boolean</code>	
<code>resources</code>	<code>object</code>	
<code>storageName</code>	<code>string</code>	
<code>timeBetweenUploads</code>	<code>integer</code>	

`.spec.mysql.pxc.backup.pitr.resources`

Type

`object`

Property	Type	Description
<code>limits</code>	<code>object</code>	

Property	Type	Description
<code>requests</code>	<code>object</code>	

`.spec.mysql.pxc.backup.pitr.resources.limits`

Type

`object`

Property	Type	Description
<code>cpu</code>	<code>string</code>	
<code>ephemeral-storage</code>	<code>string</code>	
<code>memory</code>	<code>string</code>	

`.spec.mysql.pxc.backup.pitr.resources.requests`

Type

`object`

Property	Type	Description
<code>cpu</code>	<code>string</code>	
<code>ephemeral-storage</code>	<code>string</code>	
<code>memory</code>	<code>string</code>	

`.spec.mysql.pxc.backup.schedule`

Type

`array`

.spec.mysql.pxc.backup.schedule[]

Type

object

Property	Type	Description
keep	integer	
name	string	
schedule	string	
storageName	string	

.spec.mysql.pxc.backup.storages

Type

object

.spec.mysql.pxc.clusterReplication

Type

object

Property	Type	Description
enableSemiSync	boolean	
enabled	boolean	
isReplica	boolean	
nodePortEnabled	boolean	
peerEntryHost	string	

Property	Type	Description
peerEntryPort	integer	

.spec.mysql.pxc.dcDesc

Type

object

Property	Type	Description
enableSemiSync	boolean	
extraProviderOptions	string	
id	string	
ipAddr	array	
isReplica	boolean	
name	string	
nodePortEnabled	boolean	
nodeWeight	integer	
sourceDcList	array	
sstHosts	array	
sstPort	integer	

.spec.mysql.pxc.dcDesc.ipAddr

Type

array

.spec.mysql.pxc.dcDesc.ipAddrs[]

Type

string

.spec.mysql.pxc.dcDesc.sourceDcList

Type

array

.spec.mysql.pxc.dcDesc.sourceDcList[]

Type

string

.spec.mysql.pxc.dcDesc.sstHosts

Type

array

.spec.mysql.pxc.dcDesc.sstHosts[]

Type

string

.spec.mysql.pxc.haproxy

Type

object

Property	Type	Description
affinity	object	

Property	Type	Description
annotations	object	
configuration	string	
containerSecurityContext	object	SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.
enableProbes	boolean	
enabled	boolean	
externalTrafficPolicy	string	for backwards compat
forceUnsafeBootstrap	boolean	
gracePeriod	integer	
image	string	
imagePullPolicy	string	PullPolicy describes a policy for if/when to pull a container image
imagePullSecrets	array	
labels	object	
livenessDelaySec	integer	

Property	Type	Description
<code>livenessProbes</code>	<code>object</code>	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
<code>loadBalancerSourceRanges</code>	<code>array</code>	
<code>nodeSelector</code>	<code>object</code>	
<code>podDisruptionBudget</code>	<code>object</code>	
<code>podSecurityContext</code>	<code>object</code>	PodSecurityContext holds pod-level security attributes and common container settings. Some fields are also present in container.securityContext. Field values of container.securityContext take precedence over field values of PodSecurityContext.
<code>priorityClassName</code>	<code>string</code>	
<code>readinessDelaySec</code>	<code>integer</code>	
<code>readinessProbes</code>	<code>object</code>	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
<code>replicasExternalTrafficPolicy</code>	<code>string</code>	for backwards compat

Property	Type	Description
replicasServiceType	string	Service Type string describes ingress methods for a service
resources	object	
schedulerName	string	
serverMaxConnections	integer	
serviceAccountName	string	
serviceAnnotations	object	Service Type string describes ingress methods for a service
serviceType	string	
sidecarResources	object	
size	integer	
sslInternalSecretName	string	
sslSecretName	string	
startupProbe	object	
tolerations	array	
vaultSecretName	string	
volumeSpec	object	
writeNode	string	

.spec.mysql.pxc.haproxy.affinity

Type

object

Property	Type	Description
advanced	object	Affinity is a group of affinity scheduling rules.
antiAffinityTopologyKey	string	

.spec.mysql.pxc.haproxy.affinity.advanced

Description

Affinity is a group of affinity scheduling rules.

Type

object

Property	Type	Description
nodeAffinity	object	Describes node affinity scheduling rules for the pod.
podAffinity	object	Describes pod affinity scheduling rules (e.g. co-locate this pod in the same node, zone, etc. as some other pod(s)).
podAntiAffinity	object	Describes pod anti-affinity scheduling rules (e.g. avoid putting this pod in the same node, zone, etc. as some other pod(s)).

.spec.mysql.pxc.haproxy.affinity.advanced.nodeAffinity

Description

Describes node affinity scheduling rules for the pod.

Type

object

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution affinity expressions, etc.) compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest

Property	Type	Description
		sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>object</code>	If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to an update), the system may or may not try to eventually evict the pod from its node.

`.spec.mysql.pxc.haproxy.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, `requiredDuringScheduling` affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding `matchExpressions`; the node(s) with the highest sum are the most preferred.

Type

`array`

`.spec.mysql.pxc.haproxy.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

An empty preferred scheduling term matches all objects with implicit weight 0 (i.e. it's a no-op). A null preferred scheduling term matches no objects (i.e. is also a no-op).

Type

object

Required

preference weight

Property	Type	Description
preference	object	A node selector term, associated with the corresponding weight.
weight	integer	Weight associated with matching the corresponding nodeSelectorTerm, in the range 1-100.

`.spec.mysql.pxc.haproxy.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference`

Description

A node selector term, associated with the corresponding weight.

Type

object

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	A list of node selector requirements by node's labels.
<code>matchFields</code>	<code>array</code>	A list of node selector requirements by node's fields.

`.spec.mysql.pxc.haproxy.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions`

Description

A list of node selector requirements by node's labels.

Type

`array`

`.spec.mysql.pxc.haproxy.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[]`

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

`key`

`operator`

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.mysql.pxc.haproxy.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.pxc.haproxy.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].prefer

ence.matchExpressions[].values[]

Type

string

.spec.mysql.pxc.haproxy.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields

Description

A list of node selector requirements by node's fields.

Type

array

.spec.mysql.pxc.haproxy.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[]

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	The label key that the selector applies to.

Property	Type	Description
<code>operator</code>	<code>string</code>	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
<code>values</code>	<code>array</code>	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

`.spec.mysql.pxc.haproxy.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values`

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.mysql.pxc.haproxy.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values[]`

Type

`string`

`.spec.mysql.pxc.haproxy.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution`

Description

If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to an update), the system may or may not try to eventually evict the pod from its node.

Type

`object`

Required

`nodeSelectorTerms`

Property	Type	Description
<code>nodeSelectorTerms</code>	<code>array</code>	Required. A list of node selector terms. The terms are ORed.

`.spec.mysql.pxc.haproxy.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms`

Description

Required. A list of node selector terms. The terms are ORed.

Type

`array`

`.spec.mysql.pxc.haproxy.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSel`

nodeSelectorTerms[]

Description

A null or empty node selector term matches no objects. The requirements of them are ANDed. The TopologySelectorTerm type implements a subset of the NodeSelectorTerm.

Type

object

Property	Type	Description
matchExpressions	array	A list of node selector requirements by node's labels.
matchFields	array	A list of node selector requirements by node's fields.

.spec.mysql.pxc.haproxy.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions

Description

A list of node selector requirements by node's labels.

Type

array

.spec.mysql.pxc.haproxy.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[]

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.mysql.pxc.haproxy.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.pxc.haproxy.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values[]

Type

string

.spec.mysql.pxc.haproxy.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields

Description

A list of node selector requirements by node's fields.

Type

array

.spec.mysql.pxc.haproxy.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[]

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
<code>key</code>	<code>string</code>	The label key that the selector applies to.
<code>operator</code>	<code>string</code>	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
<code>values</code>	<code>array</code>	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

`.spec.mysql.pxc.haproxy.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values`

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.mysql.pxc.haproxy.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSel`

ectorTerms[].matchFields[].values[]

Type

string

.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity

Description

Describes pod affinity scheduling rules (e.g. co-locate this pod in the same node, zone, etc. as some other pod(s)).

Type

object

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution, affinity expressions, etc.), compute a sum by iterating through the elements of this field and

Property	Type	Description
		adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	If the affinity requirement: specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirement: specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

`.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

`array`

`.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Type

`object`

Required

`podAffinityTerm``weight`

Property	Type	Description
<code>podAffinityTerm</code>	<code>object</code>	Required. A pod affinity term, associated with the corresponding weight.

Property	Type	Description
<code>weight</code>	<code>integer</code>	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

`.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm`

Description

Required. A pod affinity term, associated with the corresponding weight.

Type

`object`

Required

`topologyKey`

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be

Property	Type	Description
		ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Property	Type	Description
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffi

inityTerm.labelSelector.matchExpressions[].values[]

Type

string

.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]

Type

string

.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]

Type

string

`.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffi

affinityTerm.namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]

Type

string

.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

string

.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Type

array

.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

Defines a set of pods (namely those matching the `labelSelector` relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key `<topologyKey>` matches that of any node on which a pod of the set of pods is running

Type

object

Required

topologyKey

Property	Type	Description
<code>labelSelector</code>	object	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	array	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>mismatchLabelKeys</code>	array	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The

Property	Type	Description
		keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both <code>mismatchLabelKeys</code> and <code>labelSelector</code>. Also, <code>mismatchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. This is a beta field and requires enabling <code>MatchLabelKeysInPodAffinity</code> feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the <code>namespaces</code> field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector (<code>{}</code>) matches all namespaces.
<code>namespaces</code>	<code>array</code>	<code>namespaces</code> specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by <code>namespaceSelector</code>. null or empty namespaces list and null <code>namespaceSelector</code> means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the <code>labelSelector</code> in the specified namespaces, where co-

Property	Type	Description
		located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector

ector.matchExpressions

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.

Property	Type	Description
<code>values</code>	<code>array</code>	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]`

Type

`string`

`.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]

Type

string

.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatch

hLabelKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]

Type

string

.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namesp

namespaceSelector.matchExpressions[].values[]

Type

string

.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.mysql.pxc.haproxy.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]

Type

string

.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity

Description

Describes pod anti-affinity scheduling rules (e.g. avoid putting this pod in the same node, zone, etc. as some other pod(s)).

Type

object

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution, anti-affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods

Property	Type	Description
		which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity
.preferredDuringSchedulingIgnoredDuringExecution

Description

The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling anti-affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

array

.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

Description

The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Type

object

Required

podAffinityTerm

weight

Property	Type	Description
podAffinityTerm	object	Required. A pod affinity term, associated with the corresponding weight.
weight	integer	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity .preferredDuringSchedulingIgnoredDuringExecution[].pod AffinityTerm

Description

Required. A pod affinity term, associated with the corresponding weight.

Type

object

Required

topologyKey

Property	Type	Description
labelSelector	object	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
matchLabelKeys	array	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key in (value) to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Property	Type	Description
<code>mismatchLabelKeys</code>	array	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>namespaceSelector</code>	object	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
<code>namespaces</code>	array	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Property	Type	Description
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the <code>labelSelector</code> in the specified namespaces, where co-located is defined as running on a node whose value of the label with key <code>topologyKey</code> matches that of any node on which any of the selected pods is running. Empty <code>topologyKey</code> is not allowed.

`.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this `PodAffinityTerm` matches with no Pods.

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.

Property	Type	Description
<code>values</code>	<code>array</code>	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

`array`

.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]

Type

`string`

.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity .preferredDuringSchedulingIgnoredDuringExecution[].pod AffinityTerm.matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity .preferredDuringSchedulingIgnoredDuringExecution[].pod AffinityTerm.matchLabelKeys[]

Type

string

.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity .preferredDuringSchedulingIgnoredDuringExecution[].pod

AffinityTerm.mismatchLabelKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

**.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity
.preferredDuringSchedulingIgnoredDuringExecution[].pod
AffinityTerm.mismatchLabelKeys[]**

Type

string

**.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity
.preferredDuringSchedulingIgnoredDuringExecution[].pod
AffinityTerm.namespaceSelector**

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity .preferredDuringSchedulingIgnoredDuringExecution[].pod AffinityTerm.namespaceSelector.matchExpressions

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity .preferredDuringSchedulingIgnoredDuringExecution[].pod AffinityTerm.namespaceSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

**.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity
.preferredDuringSchedulingIgnoredDuringExecution[].pod
AffinityTerm.namespaceSelector.matchExpressions[].valu
es**

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

**.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity
.preferredDuringSchedulingIgnoredDuringExecution[].pod
AffinityTerm.namespaceSelector.matchExpressions[].valu
es[]**

Type

string

**.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity
.preferredDuringSchedulingIgnoredDuringExecution[].pod
AffinityTerm.namespaceSelector.matchLabels**

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

**.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity
.preferredDuringSchedulingIgnoredDuringExecution[].pod
AffinityTerm.namespaces**

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity .preferredDuringSchedulingIgnoredDuringExecution[].pod AffinityTerm.namespaces[]

Type

string

.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity .requiredDuringSchedulingIgnoredDuringExecution

Description

If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Type

array

.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity .requiredDuringSchedulingIgnoredDuringExecution[]

Description

Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running

Type

object

Required

topologyKey

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field

Property	Type	Description
		and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity`
`.requiredDuringSchedulingIgnoredDuringExecution[].label`

Selector

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity .requiredDuringSchedulingIgnoredDuringExecution[].label Selector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity .requiredDuringSchedulingIgnoredDuringExecution[].label`

Selector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity .requiredDuringSchedulingIgnoredDuringExecution[].label Selector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This

array is replaced during a strategic merge patch.

Type

array

**.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity
.requiredDuringSchedulingIgnoredDuringExecution[].label
Selector.matchExpressions[].values[]**

Type

string

**.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity
.requiredDuringSchedulingIgnoredDuringExecution[].label
Selector.matchLabels**

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

**.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity
.requiredDuringSchedulingIgnoredDuringExecution[].mat
chLabelKeys**

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti)

affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both `matchLabelKeys` and `labelSelector`. Also, `matchLabelKeys` cannot be set when `labelSelector` isn't set. This is a beta field and requires enabling `MatchLabelKeysInPodAffinity` feature gate (enabled by default).

Type

array

**`.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity
.requiredDuringSchedulingIgnoredDuringExecution[].mat
chLabelKeys[]`**

Type

string

**`.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity
.requiredDuringSchedulingIgnoredDuringExecution[].mis
matchLabelKeys`**

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both `mismatchLabelKeys` and `labelSelector`. Also, `mismatchLabelKeys` cannot be set when `labelSelector` isn't set. This is a beta field and requires enabling `MatchLabelKeysInPodAffinity` feature gate (enabled by default).

Type

array

**`.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity
.requiredDuringSchedulingIgnoredDuringExecution[].mis`**

matchLabelKeys[]

Type

string

.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity .requiredDuringSchedulingIgnoredDuringExecution[].names paceSelector

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity .requiredDuringSchedulingIgnoredDuringExecution[].names paceSelector

espaceSelector.matchExpressions

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity .requiredDuringSchedulingIgnoredDuringExecution[].name espaceSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.

Property	Type	Description
<code>values</code>	<code>array</code>	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]`

Type

`string`

`.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity .requiredDuringSchedulingIgnoredDuringExecution[].namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.mysql.pxc.haproxy.affinity.advanced.podAntiAffinity .requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]

Type

string

.spec.mysql.pxc.haproxy.annotations

Type

object

.spec.mysql.pxc.haproxy.containerSecurityContext

Description

SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.

Type

object

Property	Type	Description
<code>allowPrivilegeEscalation</code>	<code>boolean</code>	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the <code>no_new_privs</code> flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: 1. run as Privileged 2. has CAP_SYS_ADMIN Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>appArmorProfile</code>	<code>object</code>	appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>capabilities</code>	<code>object</code>	The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when <code>spec.os.name</code> is windows.

Property	Type	Description
<code>privileged</code>	<code>boolean</code>	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.
<code>procMount</code>	<code>string</code>	procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows.
<code>readOnlyRootFilesystem</code>	<code>boolean</code>	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
<code>runAsGroup</code>	<code>integer</code>	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
<code>runAsNonRoot</code>	<code>boolean</code>	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
<code>runAsUser</code>	<code>integer</code>	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
<code>seLinuxOptions</code>	<code>object</code>	The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
<code>seccompProfile</code>	<code>object</code>	The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.
<code>windowsOptions</code>	<code>object</code>	The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

`.spec.mysql.pxc.haproxy.containerSecurityContext.appArmorProfile`

Description

appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
<code>type</code>	<code>string</code>	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

`.spec.mysql.pxc.haproxy.containerSecurityContext.capabilities`

Description

The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when `spec.os.name` is windows.

Type

`object`

Property	Type	Description
<code>add</code>	<code>array</code>	Added capabilities
<code>drop</code>	<code>array</code>	Removed capabilities

`.spec.mysql.pxc.haproxy.containerSecurityContext.capabilities.add`

Description

Added capabilities

Type

array

`.spec.mysql.pxc.haproxy.containerSecurityContext.capabilities.add[]`

Description

Capability represent POSIX capabilities type

Type

string

`.spec.mysql.pxc.haproxy.containerSecurityContext.capabilities.drop`

Description

Removed capabilities

Type

array

`.spec.mysql.pxc.haproxy.containerSecurityContext.capabilities.drop[]`

Description

Capability represent POSIX capabilities type

Type

string

.spec.mysql.pxc.haproxy.containerSecurityContext.seLinuxOptions

Description

The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Type

object

Property	Type	Description
level	string	Level is SELinux level label that applies to the container.
role	string	Role is a SELinux role label that applies to the container.
type	string	Type is a SELinux type label that applies to the container.
user	string	User is a SELinux user label that applies to the container.

.spec.mysql.pxc.haproxy.containerSecurityContext.seccompProfile

Description

The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.

Type

object

Required

type

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.
type	string	type indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied.

.spec.mysql.pxc.haproxy.containerSecurityContext.windowsOptions

Description

The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the

value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the

Property	Type	Description
		value specified in SecurityContext takes precedence.

`.spec.mysql.pxc.haproxy.imagePullSecrets`

Type

array

`.spec.mysql.pxc.haproxy.imagePullSecrets[]`

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

`.spec.mysql.pxc.haproxy.labels`

Type

object

.spec.mysql.pxc.haproxy.livenessProbes

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
<code>exec</code>	object	Exec specifies a command to execute in the
<code>failureThreshold</code>	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Default value is 3. Minimum value is 1.
<code>grpc</code>	object	GRPC specifies a GRPC HealthCheckRequest to
<code>httpGet</code>	object	HTTPGet specifies an HTTP GET request to
<code>initialDelaySeconds</code>	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes
<code>periodSeconds</code>	integer	How often (in seconds) to perform the probe. Default is 10 seconds. Minimum value is 1.

Property	Type	Description
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies a connection to a TCP endpoint.
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod needs to gracefully shutdown before it is terminated. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. A value greater than 0 indicates the grace period should be used. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. spec.terminationGracePeriodSeconds is used if specified.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More information: https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes

.spec.mysql.pxc.haproxy.livenessProbes.exec

Description

Exec specifies a command to execute in the container.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.mysql.pxc.haproxy.livenessProbes.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

.spec.mysql.pxc.haproxy.livenessProbes.exec.command[]

Type

string

.spec.mysql.pxc.haproxy.livenessProbes.grpc

Description

GRPC specifies a GRPC HealthCheckRequest.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

.spec.mysql.pxc.haproxy.livenessProbes.httpGet

Description

HTTPGet specifies an HTTP GET request to perform.

Type

object

Required

port

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.mysql.pxc.haproxy.livenessProbes.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.mysql.pxc.haproxy.livenessProbes.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.mysql.pxc.haproxy.livenessProbes.tcpSocket

Description

TCPSocket specifies a connection to a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

.spec.mysql.pxc.haproxy.loadBalancerSourceRanges

Type

`array`

.spec.mysql.pxc.haproxy.loadBalancerSourceRanges[]

Type

`string`

.spec.mysql.pxc.haproxy.nodeSelector

Type

`object`

.spec.mysql.pxc.haproxy.podDisruptionBudget

Type

`object`

Property	Type	Description
<code>maxUnavailable</code>		
<code>minAvailable</code>		

.spec.mysql.pxc.haproxy.podSecurityContext

Description

PodSecurityContext holds pod-level security attributes and common container settings. Some fields are also present in container.securityContext. Field values of container.securityContext take precedence over field values of PodSecurityContext.

Type

object

Property	Type	Description
appArmorProfile	object	appArmorProfile is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.
fsGroup	integer	A special supplemental group that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod: 1. The owning GID will be the FSGroup 2. The setgid bit is set (new files created in the volume will be owned by FSGroup) 3. The permission bits are OR'd with rw-rw-- -- If unset, the Kubelet will not modify the ownership and permissions of any volume. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
<code>fsGroupChangePolicy</code>	<code>string</code>	<code>fsGroupChangePolicy</code> defines behavior of changing ownership and permission of the volume before being exposed inside Pod. This field will only apply to volume types which support <code>fsGroup</code> based ownership(and permissions). It will have no effect on ephemeral volume types such as: <code>secret</code>, <code>configmaps</code> and <code>emptydir</code>. Valid values are "OnRootMismatch" and "Always". If not specified, "Always" is used. Note that this field cannot be set when <code>spec.os.name</code> is <code>windows</code>.
<code>runAsGroup</code>	<code>integer</code>	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in <code>SecurityContext</code>. If set in both <code>SecurityContext</code> and <code>PodSecurityContext</code>, the value specified in <code>SecurityContext</code> takes precedence for that container. Note that this field cannot be set when <code>spec.os.name</code> is <code>windows</code>.
<code>runAsNonRoot</code>	<code>boolean</code>	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in <code>SecurityContext</code>. If set in both <code>SecurityContext</code> and <code>PodSecurityContext</code>,

Property	Type	Description
		the value specified in SecurityContext takes precedence.
<code>runAsUser</code>	<code>integer</code>	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
<code>seLinuxChangePolicy</code>	<code>string</code>	<code>seLinuxChangePolicy</code> defines how the container's SELinux label is applied to all volumes used by the Pod. It has no effect on nodes that do not support SELinux or to volumes does not support SELinux. Valid values are "MountOption" and "Recursive". "Recursive" means relabeling of all files on all Pod volumes by the container runtime. This may be slow for large volumes, but allows mixing privileged and unprivileged Pods sharing the same volume on the same node. "MountOption" mounts all eligible Pod volumes with <code>-o context</code> mount option. This requires all Pods that share the same volume to use the same SELinux label. It is not possible to share the same volume among privileged and unprivileged Pods.

Property	Type	Description
		Eligible volumes are in-tree FibreChannel and iSCSI volumes, and all CSI volumes whose CSI driver announces SELinux support by setting spec.seLinuxMount: true in their CSIDriver instance. Other volumes are always re-labelled recursively. "MountOption" value is allowed only when SELinuxMount feature gate is enabled. If not specified and SELinuxMount feature gate is enabled, "MountOption" is used. If not specified and SELinuxMount feature gate is disabled, "MountOption" is used for ReadWriteOncePod volumes and "Recursive" for all other volumes. This field affects only Pods that have SELinux label set, either in PodSecurityContext or in SecurityContext of all containers. All Pods that use the same volume should use the same seLinuxChangePolicy, otherwise some pods can get stuck in ContainerCreating state. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes

Property	Type	Description
		precedence for that container. Note that this field cannot be set when spec.os.name is windows.
seccompProfile	object	The seccomp options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.
supplementalGroups	array	A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.
supplementalGroupsPolicy	string	Defines how supplemental groups of the first container processes are calculated. Valid values are "Merge" and "Strict". If not specified, "Merge" is used. (Alpha) Using the field requires the SupplementalGroupsPolicy feature gate to be enabled and the container

Property	Type	Description
		runtime must implement support for this feature. Note that this field cannot be set when spec.os.name is windows.
<code>sysctls</code>	<code>array</code>	Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.
<code>windowsOptions</code>	<code>object</code>	The Windows specific settings applied to all containers. If unspecified, the options within a container's SecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

`.spec.mysql.pxc.haproxy.podSecurityContext.appArmorProfile`

Description

appArmorProfile is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
<code>type</code>	<code>string</code>	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

`.spec.mysql.pxc.haproxy.podSecurityContext.seLinuxOptions`

Description

The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Property	Type	Description
<code>level</code>	<code>string</code>	Level is SELinux level label that applies to the container.

Property	Type	Description
<code>role</code>	<code>string</code>	Role is a SELinux role label that applies to the container.
<code>type</code>	<code>string</code>	Type is a SELinux type label that applies to the container.
<code>user</code>	<code>string</code>	User is a SELinux user label that applies to the container.

`.spec.mysql.pxc.haproxy.podSecurityContext.seccompProfile`

Description

The seccomp options to use by the containers in this pod. Note that this field cannot be set when `spec.os.name` is `windows`.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.

Property	Type	Description
<code>type</code>	<code>string</code>	type indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied.

`.spec.mysql.pxc.haproxy.podSecurityContext.supplementalGroups`

Description

A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.

Type

`array`

`.spec.mysql.pxc.haproxy.podSecurityContext.supplementalGroups[]`

Type

`integer`

`.spec.mysql.pxc.haproxy.podSecurityContext.sysctls`

Description

Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.

Type

array

.spec.mysql.pxc.haproxy.podSecurityContext.sysctls[]

Description

Sysctl defines a kernel parameter to be set

Type

object

Required

name value

Property	Type	Description
name	string	Name of a property to set
value	string	Value of a property to set

.spec.mysql.pxc.haproxy.podSecurityContext.windowsOptions

Description

The Windows specific settings applied to all containers. If unspecified, the options within a container's SecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
gmsaCredentialSpec	string	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa [↗]) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
gmsaCredentialSpecName	string	GMSACredentialSpecName is the name of the GMSA credential spec to use.
hostProcess	boolean	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.

Property	Type	Description
runAsUserName	string	The UserName in Windows to run the entryptoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

.spec.mysql.pxc.haproxy.readinessProbes

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	Exec specifies a command to execute in the
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Default value is 3. Minimum value is 1.
grpc	object	GRPC specifies a GRPC HealthCheckRequest to send to the service.

Property	Type	Description
<code>httpGet</code>	<code>object</code>	HTTPGet specifies an HTTP GET request to
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/lifecycle#container-probes
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe. Default is 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe to be considered successful after having failed. Default is 1. Must be 1 for liveness and startup. Minimum value is 1.
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies a connection to a TCP port.
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod needs to gracefully shutdown before it is killed. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. It specifies that the pod needs to terminate gracefully within this time by default. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. If the value is zero, it indicates that the pod should be terminated immediately via the SIGKILL signal.

Property	Type	Description
		(no opportunity to shut down). This is a beta requires enabling ProbeTerminationGracePeriod. Minimum value is 1. spec.terminationGracePeriodSeconds is use
timeoutSeconds	integer	Number of seconds after which the probe timeout. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/lifecycle#container-probes

.spec.mysql.pxc.haproxy.readinessProbes.exec

Description

Exec specifies a command to execute in the container.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.mysql.pxc.haproxy.readinessProbes.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

.spec.mysql.pxc.haproxy.readinessProbes.exec.command



Type

string

.spec.mysql.pxc.haproxy.readinessProbes.grpc

Description

GRPC specifies a GRPC HealthCheckRequest.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see

Property	Type	Description
		https://github.com/grpc/grpc/blob/master/doc/health-checking.md ). If this is not specified, the default behavior is defined by gRPC.

.spec.mysql.pxc.haproxy.readinessProbes.httpGet

Description

HTTPGet specifies an HTTP GET request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

Property	Type	Description
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.mysql.pxc.haproxy.readinessProbes.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.mysql.pxc.haproxy.readinessProbes.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

- `name`
- `value`

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.mysql.pxc.haproxy.readinessProbes.tcpSocket`

Description

TCPSocket specifies a connection to a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.mysql.pxc.haproxy.resources`

Type

object

Property	Type	Description
limits	object	
requests	object	

`.spec.mysql.pxc.haproxy.resources.limits`

Type

`object`

Property	Type	Description
<code>cpu</code>	<code>string</code>	
<code>ephemeral-storage</code>	<code>string</code>	
<code>memory</code>	<code>string</code>	

`.spec.mysql.pxc.haproxy.resources.requests`

Type

`object`

Property	Type	Description
<code>cpu</code>	<code>string</code>	
<code>ephemeral-storage</code>	<code>string</code>	
<code>memory</code>	<code>string</code>	

`.spec.mysql.pxc.haproxy.serviceAnnotations`

Type

`object`

`.spec.mysql.pxc.haproxy.sidecarResources`

Type

`object`

Property	Type	Description
limits	object	
requests	object	

.spec.mysql.pxc.haproxy.sidecarResources.limits

Type

object

Property	Type	Description
cpu	string	
ephemeral-storage	string	
memory	string	

.spec.mysql.pxc.haproxy.sidecarResources.requests

Type

object

Property	Type	Description
cpu	string	
ephemeral-storage	string	
memory	string	

.spec.mysql.pxc.haproxy.startupProbe

Type

object

Property	Type	Description
failureThreshold	integer	
initialDelaySeconds	integer	
periodSeconds	integer	

.spec.mysql.pxc.haproxy.tolerations

Type

array

.spec.mysql.pxc.haproxy.tolerations[]

Description

The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.

Type

object

Property	Type	Description
effect	string	Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.
key	string	Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is

Property	Type	Description
		empty, operator must be Exists; this combination means to match all values and all keys.
<code>operator</code>	<code>string</code>	Operator represents a key's relationship to the value. Valid operators are Exists and Equal. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category.
<code>tolerationSeconds</code>	<code>integer</code>	TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is not set, which means tolerate the taint forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.
<code>value</code>	<code>string</code>	Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

`.spec.mysql.pxc.haproxy.volumeSpec`

Type

`object`

Property	Type	Description
<code>emptyDir</code>	<code>object</code>	EmptyDir to use as data volume for mysql. EmptyDir represents a temporary directory that

Property	Type	Description
		shares a pod's lifetime.
<code>hostPath</code>	<code>object</code>	HostPath to use as data volume for mysql. HostPath represents a pre-existing file or directory on the host machine that is directly exposed to the container.
<code>persistentVolumeClaim</code>	<code>object</code>	PersistentVolumeClaim to specify PVC spec for the volume for mysql data. It has the highest level of precedence, followed by HostPath and EmptyDir. And represents the PVC specification.

`.spec.mysql.pxc.haproxy.volumeSpec.emptyDir`

Description

EmptyDir to use as data volume for mysql. EmptyDir represents a temporary directory that shares a pod's lifetime.

Type

`object`

Property	Type	Description
<code>medium</code>	<code>string</code>	medium represents what type of storage medium should back this directory. The default is "" which means to use the node's default medium. Must be an empty string (default) or Memory. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir

Property	Type	Description
<code>sizeLimit</code>		sizeLimit is the total amount of local storage required for this EmptyDir volume. The size limit is also applicable for memory medium. The maximum usage on memory medium EmptyDir would be the minimum value between the SizeLimit specified here and the sum of memory limits of all containers in a pod. The default is nil which means that the limit is undefined. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir

.spec.mysql.pxc.haproxy.volumeSpec.hostPath

Description

HostPath to use as data volume for mysql. HostPath represents a pre-existing file or directory on the host machine that is directly exposed to the container.

Type

`object`

Required

`path`

Property	Type	Description
<code>path</code>	<code>string</code>	path of the directory on the host. If the path is a symlink, it will follow the link to the real path. More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath
<code>type</code>	<code>string</code>	type for HostPath Volume Defaults to "" More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath

.spec.mysql.pxc.haproxy.volumeSpec.persistentVolumeClaim

Description

PersistentVolumeClaim to specify PVC spec for the volume for mysql data. It has the highest level of precedence, followed by HostPath and EmptyDir. And represents the PVC specification.

Type

object

Property	Type	Description
<code>accessModes</code>	array	accessModes contains the desired access modes volume should have. More info: https://kubernetes.io/docs/concepts/storage/persistentvolumes#access-modes-1
<code>dataSource</code>	object	dataSource field can be used to specify either: • An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) • An existing PVC (PersistentVolumeClaim) If the provisioner or an external controller can support the specified data source, it will create a new volume based on the contents of the specified data source. When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If the namespace is specified, then dataSourceRef will not be copied to dataSource.

Property	Type	Description
dataSourceRef	object	dataSourceRef specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a PersistentVolumeClaim object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will require the functionality of the dataSource field and as such both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in dataSourceRef, dataSource isn't set to the same value and must be empty. There are three important differences between dataSource and dataSourceRef • While dataSource only allows two specific type objects, dataSourceRef allows any non-core object as well as PersistentVolumeClaim objects. • While dataSource ignores disallowed values (dropping them), dataSourceRef preserves all values, and generates an error if a disallowed value is specified. • While dataSource only allows local objects, dataSourceRef allows objects in any namespace (Beta) Using this field requires the AnyVolumeDataSource feature gate to be enabled (Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

Property	Type	Description
resources	object	resources represents the minimum resources the volume should have. If RecoverVolumeExpansionFailure feature is enable users are allowed to specify resource requirement that are lower than previous value but must still be higher than capacity recorded in the status field of claim. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources
selector	object	selector is a label query over volumes to consider binding.
storageClassName	string	storageClassName is the name of the StorageClass required by the claim. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#class-1
volumeAttributesClassName	string	volumeAttributesClassName may be used to set the VolumeAttributesClass used by this claim. If specified the CSI driver will create or update the volume with attributes defined in the corresponding VolumeAttributesClass. This has a different purpose than storageClassName, it can be changed after the claim is created. An empty string value means that VolumeAttributesClass will be applied to the claim it's not allowed to reset this field to empty string once it is set. If unspecified and the PersistentVolumeClaim is unbound, the default VolumeAttributesClass will be

Property	Type	Description
		by the persistentvolume controller if it exists. If the resource referred to by volumeAttributesClass does not exist, this PersistentVolumeClaim will be set to Pending state, as reflected by the modifyVolumeStatus field, until such as a resource exists. More info: https://kubernetes.io/docs/concepts/storage/volume-attributes-classes/ (Beta) Using this field requires the VolumeAttributesClass feature gate to be enabled (off by default).
volumeMode	string	volumeMode defines what type of volume is required by the claim. Value of Filesystem is implied when not included in claim spec.
volumeName	string	volumeName is the binding reference to the PersistentVolume backing this claim.

`.spec.mysql.pxc.haproxy.volumeSpec.persistentVolumeClaim.accessModes`

Description

accessModes contains the desired access modes the volume should have. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1>

Type

array

`.spec.mysql.pxc.haproxy.volumeSpec.persistentVolumeClaim.accessModes[]`

Type

string

.spec.mysql.pxc.haproxy.volumeSpec.persistentVolumeClaim.dataSource

Description

dataSource field can be used to specify either:

- * An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot)
- * An existing PVC (PersistentVolumeClaim)

If the provisioner or an external controller can support the specified data source, it will create a new volume based on the contents of the specified data source. When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If the namespace is specified, then dataSourceRef will not be copied to dataSource.

Type

object

Required

kind

name

Property	Type	Description
apiGroup	string	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
kind	string	Kind is the type of resource being referenced
name	string	Name is the name of resource being referenced

.spec.mysql.pxc.haproxy.volumeSpec.persistentVolumeClaim.dataSourceRef

Description

dataSourceRef specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a PersistentVolumeClaim object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replace the functionality of the dataSource field and as such if both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in dataSourceRef, dataSource isn't set to the same value and must be empty. There are three important differences between dataSource and dataSourceRef: * While dataSource only allows two specific types of objects, dataSourceRef allows any non-core object, as well as PersistentVolumeClaim objects. * While dataSource ignores disallowed values (dropping them), dataSourceRef preserves all values, and generates an error if a disallowed value is specified. * While dataSource only allows local objects, dataSourceRef allows objects in any namespaces. (Beta) Using this field requires the AnyVolumeDataSource feature gate to be enabled. (Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

Type

object

Required

kind

name

Property	Type	Description
apiGroup	string	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.

Property	Type	Description
<code>kind</code>	<code>string</code>	Kind is the type of resource being referenced
<code>name</code>	<code>string</code>	Name is the name of resource being referenced
<code>namespace</code>	<code>string</code>	Namespace is the namespace of resource being referenced Note that when a namespace is specified, a <code>gateway.networking.k8s.io/ReferenceGrant</code> object is required in the referent namespace to allow that namespace's owner to accept the reference. See the <code>ReferenceGrant</code> documentation for details. (Alpha) This field requires the <code>CrossNamespaceVolumeDataSource</code> feature gate to be enabled.

`.spec.mysql.pxc.haproxy.volumeSpec.persistentVolumeClaim.resources`

Description

`resources` represents the minimum resources the volume should have. If `RecoverVolumeExpansionFailure` feature is enabled users are allowed to specify resource requirements that are lower than previous value but must still be higher than capacity recorded in the status field of the claim. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources>

Type

`object`

Property	Type	Description
<code>limits</code>	<code>object</code>	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗
<code>requests</code>	<code>object</code>	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

`.spec.mysql.pxc.haproxy.volumeSpec.persistentVolumeClaim.resources.limits`

Description

Limits describes the maximum amount of compute resources allowed. More info:
<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.mysql.pxc.haproxy.volumeSpec.persistentVolumeClaim.resources.requests`

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an

implementation-defined value. Requests cannot exceed Limits. More info:
<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

`.spec.mysql.pxc.haproxy.volumeSpec.persistentVolumeClaim.selector`

Description

selector is a label query over volumes to consider for binding.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.pxc.haproxy.volumeSpec.persistentVolumeClaim.selector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.mysql.pxc.haproxy.volumeSpec.persistentVolumeClaim.selector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.mysql.pxc.haproxy.volumeSpec.persistentVolumeClaim.selector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.pxc.haproxy.volumeSpec.persistentVolumeClaim.selector.matchExpressions[].values[]

Type

string

.spec.mysql.pxc.haproxy.volumeSpec.persistentVolumeClaim.selector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mysql.pxc.logcollector

Type

object

Property	Type	Description
configuration	string	

Property	Type	Description
<code>containerSecurityContext</code>	<code>object</code>	SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.
<code>enabled</code>	<code>boolean</code>	
<code>image</code>	<code>string</code>	
<code>imagePullPolicy</code>	<code>string</code>	PullPolicy describes a policy for if/when to pull a container image
<code>resources</code>	<code>object</code>	

`.spec.mysql.pxc.logcollector.containerSecurityContext`

Description

SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.

Type

`object`

Property	Type	Description
<code>allowPrivilegeEscalation</code>	<code>boolean</code>	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the <code>no_new_privs</code> flag will be set on the

Property	Type	Description
		container process. AllowPrivilegeEscalation is true always when the container is: 1. run as Privileged 2. has CAP_SYS_ADMIN Note that this field cannot be set when spec.os.name is windows.
appArmorProfile	object	appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when spec.os.name is windows.
capabilities	object	The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when spec.os.name is windows.
privileged	boolean	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.
procMount	string	procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked

Property	Type	Description
		paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows.
readOnlyRootFilesystem	boolean	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

Property	Type	Description
<code>runAsUser</code>	<code>integer</code>	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
<code>seLinuxOptions</code>	<code>object</code>	The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
<code>seccompProfile</code>	<code>object</code>	The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.
<code>windowsOptions</code>	<code>object</code>	The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and

Property	Type	Description
		PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

.spec.mysql.pxc.logcollector.containerSecurityContext.appArmorProfile

Description

appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when spec.os.name is windows.

Type

object

Required

type

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
type	string	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

`.spec.mysql.pxc.logcollector.containerSecurityContext.capabilities`

Description

The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when `spec.os.name` is windows.

Type

object

Property	Type	Description
<code>add</code>	array	Added capabilities
<code>drop</code>	array	Removed capabilities

`.spec.mysql.pxc.logcollector.containerSecurityContext.capabilities.add`

Description

Added capabilities

Type

array

`.spec.mysql.pxc.logcollector.containerSecurityContext.capabilities.add[]`

Description

Capability represent POSIX capabilities type

Type

string

.spec.mysql.pxc.logcollector.containerSecurityContext.capabilities.drop**Description**

Removed capabilities

Type

array

.spec.mysql.pxc.logcollector.containerSecurityContext.capabilities.drop[]**Description**

Capability represent POSIX capabilities type

Type

string

.spec.mysql.pxc.logcollector.containerSecurityContext.seLinuxOptions**Description**

The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Type

object

Property	Type	Description
<code>level</code>	<code>string</code>	Level is SELinux level label that applies to the container.
<code>role</code>	<code>string</code>	Role is a SELinux role label that applies to the container.
<code>type</code>	<code>string</code>	Type is a SELinux type label that applies to the container.
<code>user</code>	<code>string</code>	User is a SELinux user label that applies to the container.

`.spec.mysql.pxc.logcollector.containerSecurityContext.seccompProfile`

Description

The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when `spec.os.name` is windows.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	localhostProfile indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured

Property	Type	Description
		seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.
<code>type</code>	<code>string</code>	type indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied.

`.spec.mysql.pxc.logcollector.containerSecurityContext.windowsOptions`

Description

The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

`object`

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

.spec.mysql.pxc.logcollector.resources

Type

object

Property	Type	Description
limits	object	
requests	object	

.spec.mysql.pxc.logcollector.resources.limits

Type

object

Property	Type	Description
cpu	string	
ephemeral-storage	string	
memory	string	

.spec.mysql.pxc.logcollector.resources.requests

Type

object

Property	Type	Description
cpu	string	
ephemeral-storage	string	
memory	string	

.spec.mysql.pxc.multiDcSpec

Type

object

Property	Type	Description
dcMembers	array	
enabled	boolean	
myId	integer	

.spec.mysql.pxc.multiDcSpec.dcMembers

Type

array

.spec.mysql.pxc.multiDcSpec.dcMembers[]

Type

object

Property	Type	Description
enableSemiSync	boolean	
extraProviderOptions	string	
id	string	
ipAddr	array	
isReplica	boolean	
name	string	

Property	Type	Description
<code>nodePortEnabled</code>	<code>boolean</code>	
<code>nodeWeight</code>	<code>integer</code>	
<code>sourceDcList</code>	<code>array</code>	
<code>sstHosts</code>	<code>array</code>	
<code>sstPort</code>	<code>integer</code>	

`.spec.mysql.pxc.multiDcSpec.dcMembers[].ipAddr`

Type

`array`

`.spec.mysql.pxc.multiDcSpec.dcMembers[].ipAddr[]`

Type

`string`

`.spec.mysql.pxc.multiDcSpec.dcMembers[].sourceDcList`

Type

`array`

**`.spec.mysql.pxc.multiDcSpec.dcMembers[].sourceDcList[
]`**

Type

`string`

`.spec.mysql.pxc.multiDcSpec.dcMembers[].sstHosts`

Type

array

`.spec.mysql.pxc.multiDcSpec.dcMembers[].sstHosts[]`

Type

string

`.spec.mysql.pxc.mysqlExporter`

Type

object

Property	Type	Description
<code>enabled</code>	<code>boolean</code>	
<code>image</code>	<code>string</code>	
<code>resources</code>	<code>object</code>	

`.spec.mysql.pxc.mysqlExporter.resources`

Type

object

Property	Type	Description
<code>limits</code>	<code>object</code>	
<code>requests</code>	<code>object</code>	

.spec.mysql.pxc.mysqlExporter.resources.limits

Type

object

Property	Type	Description
cpu	string	
ephemeral-storage	string	
memory	string	

.spec.mysql.pxc.mysqlExporter.resources.requests

Type

object

Property	Type	Description
cpu	string	
ephemeral-storage	string	
memory	string	

.spec.mysql.pxc.proxysql

Type

object

Property	Type	Description
affinity	object	
annotations	object	

Property	Type	Description
<code>configuration</code>	<code>string</code>	
<code>containerSecurityContext</code>	<code>object</code>	SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.
<code>enableProbes</code>	<code>boolean</code>	
<code>enabled</code>	<code>boolean</code>	
<code>externalTrafficPolicy</code>	<code>string</code>	for backwards compat
<code>forceUnsafeBootstrap</code>	<code>boolean</code>	
<code>gracePeriod</code>	<code>integer</code>	
<code>image</code>	<code>string</code>	
<code>imagePullPolicy</code>	<code>string</code>	PullPolicy describes a policy for if/when to pull a container image
<code>imagePullSecrets</code>	<code>array</code>	
<code>labels</code>	<code>object</code>	
<code>livenessDelaySec</code>	<code>integer</code>	

Property	Type	Description
<code>livenessProbes</code>	<code>object</code>	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
<code>loadBalancerSourceRanges</code>	<code>array</code>	
<code>nodeSelector</code>	<code>object</code>	
<code>podDisruptionBudget</code>	<code>object</code>	
<code>podSecurityContext</code>	<code>object</code>	PodSecurityContext holds pod-level security attributes and common container settings. Some fields are also present in container.securityContext. Field values of container.securityContext take precedence over field values of PodSecurityContext.
<code>priorityClassName</code>	<code>string</code>	
<code>readinessDelaySec</code>	<code>integer</code>	
<code>readinessProbes</code>	<code>object</code>	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
<code>replicasExternalTrafficPolicy</code>	<code>string</code>	for backwards compat

Property	Type	Description
replicasServiceType	string	Service Type string describes ingress methods for a service
resources	object	
schedulerName	string	
serverMaxConnections	integer	
serviceAccountName	string	
serviceAnnotations	object	Service Type string describes ingress methods for a service
serviceType	string	
sidecarResources	object	
size	integer	
sslInternalSecretName	string	
sslSecretName	string	
startupProbe	object	
tolerations	array	
vaultSecretName	string	
volumeSpec	object	
writeNode	string	

.spec.mysql.pxc.proxysql.affinity

Type

object

Property	Type	Description
advanced	object	Affinity is a group of affinity scheduling rules.
antiAffinityTopologyKey	string	

.spec.mysql.pxc.proxysql.affinity.advanced

Description

Affinity is a group of affinity scheduling rules.

Type

object

Property	Type	Description
nodeAffinity	object	Describes node affinity scheduling rules for the pod.
podAffinity	object	Describes pod affinity scheduling rules (e.g. co-locate this pod in the same node, zone, etc. as some other pod(s)).
podAntiAffinity	object	Describes pod anti-affinity scheduling rules (e.g. avoid putting this pod in the same node, zone, etc. as some other pod(s)).

.spec.mysql.pxc.proxysql.affinity.advanced.nodeAffinity

Description

Describes node affinity scheduling rules for the pod.

Type

object

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution affinity expressions, etc.) compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest

Property	Type	Description
		sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>object</code>	If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to an update), the system may or may not try to eventually evict the pod from its node.

`.spec.mysql.pxc.proxysql.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, `requiredDuringScheduling` affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding `matchExpressions`; the node(s) with the highest sum are the most preferred.

Type

`array`

`.spec.mysql.pxc.proxysql.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

An empty preferred scheduling term matches all objects with implicit weight 0 (i.e. it's a no-op). A null preferred scheduling term matches no objects (i.e. is also a no-op).

Type

`object`

Required

`preference``weight`

Property	Type	Description
<code>preference</code>	<code>object</code>	A node selector term, associated with the corresponding weight.
<code>weight</code>	<code>integer</code>	Weight associated with matching the corresponding nodeSelectorTerm, in the range 1-100.

`.spec.mysql.pxc.proxysql.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference`

Description

A node selector term, associated with the corresponding weight.

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	A list of node selector requirements by node's labels.
<code>matchFields</code>	<code>array</code>	A list of node selector requirements by node's fields.

`.spec.mysql.pxc.proxysql.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions`

Description

A list of node selector requirements by node's labels.

Type

`array`

`.spec.mysql.pxc.proxysql.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[]`

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

`key`

`operator`

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.mysql.pxc.proxysql.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.pxc.proxysql.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference

rence.matchExpressions[].values[]

Type

string

.spec.mysql.pxc.proxysql.affinity.advanced.nodeAffinity.p referredDuringSchedulingIgnoredDuringExecution[].prefe rence.matchFields

Description

A list of node selector requirements by node's fields.

Type

array

.spec.mysql.pxc.proxysql.affinity.advanced.nodeAffinity.p referredDuringSchedulingIgnoredDuringExecution[].prefe rence.matchFields[]

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	The label key that the selector applies to.

Property	Type	Description
<code>operator</code>	<code>string</code>	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
<code>values</code>	<code>array</code>	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

`.spec.mysql.pxc.proxysql.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values`

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.mysql.pxc.proxysql.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values[]`

Type

`string`

.spec.mysql.pxc.proxysql.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to an update), the system may or may not try to eventually evict the pod from its node.

Type

object

Required

nodeSelectorTerms

Property	Type	Description
nodeSelectorTerms	array	Required. A list of node selector terms. The terms are ORed.

.spec.mysql.pxc.proxysql.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms

Description

Required. A list of node selector terms. The terms are ORed.

Type

array

.spec.mysql.pxc.proxysql.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSe

lectorTerms[]

Description

A null or empty node selector term matches no objects. The requirements of them are ANDed. The TopologySelectorTerm type implements a subset of the NodeSelectorTerm.

Type

object

Property	Type	Description
matchExpressions	array	A list of node selector requirements by node's labels.
matchFields	array	A list of node selector requirements by node's fields.

.spec.mysql.pxc.proxysql.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions

Description

A list of node selector requirements by node's labels.

Type

array

.spec.mysql.pxc.proxysql.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[]

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.mysql.pxc.proxysql.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.pxc.proxysql.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values[]

Type

string

.spec.mysql.pxc.proxysql.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields

Description

A list of node selector requirements by node's fields.

Type

array

.spec.mysql.pxc.proxysql.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[]

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.mysql.pxc.proxysql.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.pxc.proxysql.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSe

lectorTerms[].matchFields[].values[]

Type

string

.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity

Description

Describes pod affinity scheduling rules (e.g. co-locate this pod in the same node, zone, etc. as some other pod(s)).

Type

object

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution affinity expressions, etc.), compute a sum by iterating through the elements of this field and

Property	Type	Description
		adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	If the affinity requirement: specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirement: specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

`.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

`array`

`.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Type

`object`

Required

`podAffinityTerm``weight`

Property	Type	Description
<code>podAffinityTerm</code>	<code>object</code>	Required. A pod affinity term, associated with the corresponding weight.

Property	Type	Description
<code>weight</code>	<code>integer</code>	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

`.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm`

Description

Required. A pod affinity term, associated with the corresponding weight.

Type

`object`

Required

`topologyKey`

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be

Property	Type	Description
		ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Property	Type	Description
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

finityTerm.labelSelector.matchExpressions[].values[]

Type

string

.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]

Type

string

.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]

Type

string

`.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]

finityTerm.namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values
[]

Type

string

.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

finityTerm.namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

string

.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Type

array

.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

Defines a set of pods (namely those matching the `labelSelector` relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key `<topologyKey>` matches that of any node on which a pod of the set of pods is running

Type

object

Required

`topologyKey`

Property	Type	Description
<code>labelSelector</code>	object	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	array	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both <code>matchLabelKeys</code> and <code>labelSelector</code> . Also, <code>matchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. This is a beta field and requires enabling <code>MatchLabelKeysInPodAffinity</code> feature gate (enabled by default).

Property	Type	Description
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Property	Type	Description
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
<code>key</code>	<code>string</code>	key is the label key that the selector applies to.
<code>operator</code>	<code>string</code>	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.

Property	Type	Description
<code>values</code>	<code>array</code>	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]`

Type

`string`

`.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]

Type

string

.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatch

chLabelKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]

Type

string

.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespacedSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespacedSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespace

paceSelector.matchExpressions[].values[]

Type

string

**.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].names
paceSelector.matchLabels**

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

**.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].names
paces**

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

**.spec.mysql.pxc.proxysql.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].names
paces[]**

Type

string

.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity

Description

Describes pod anti-affinity scheduling rules (e.g. avoid putting this pod in the same node, zone, etc. as some other pod(s)).

Type

object

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution, anti-affinity expressions, etc.), compute a sum by iterating through the elements of this field and

Property	Type	Description
		adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

`.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling anti-affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

`array`

`.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Type

`object`

Required

`podAffinityTerm``weight`

Property	Type	Description
<code>podAffinityTerm</code>	<code>object</code>	Required. A pod affinity term, associated with the corresponding weight.

Property	Type	Description
<code>weight</code>	<code>integer</code>	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

`.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm`

Description

Required. A pod affinity term, associated with the corresponding weight.

Type

`object`

Required

`topologyKey`

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be

Property	Type	Description
		ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Property	Type	Description
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].po

`dAffinityTerm.labelSelector.matchExpressions[].values[]`

Type

string

`.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

`.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys`

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]

Type

string

.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]

Type

string

`.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].po

`dAffinityTerm.namespaceSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]`

Type

string

`.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

string

.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Type

array

```
.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.  
y.requiredDuringSchedulingIgnoredDuringExecution[]
```

Description

Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running

Type

object

Required

topologyKey

Property	Type	Description
labelSelector	object	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Property	Type	Description
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Property	Type	Description
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector (<code>{}</code>) matches all namespaces.
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]

Type

string

.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]

Type

string

.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]

Type

string

`.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

`key``operator`

Property	Type	Description
<code>key</code>	<code>string</code>	key is the label key that the selector applies to.
<code>operator</code>	<code>string</code>	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
<code>values</code>	<code>array</code>	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].na`

mespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]

Type

string

.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.mysql.pxc.proxysql.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]

Type

string

.spec.mysql.pxc.proxysql.annotations

Type

object

.spec.mysql.pxc.proxysql.containerSecurityContext

Description

SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.

Type

object

Property	Type	Description
<code>allowPrivilegeEscalation</code>	<code>boolean</code>	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the <code>no_new_privs</code> flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: 1. run as Privileged 2. has CAP_SYS_ADMIN Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>appArmorProfile</code>	<code>object</code>	<code>appArmorProfile</code> is the AppArmor options to use by this container. If set, this profile overrides the pod's <code>appArmorProfile</code>. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>capabilities</code>	<code>object</code>	The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>privileged</code>	<code>boolean</code>	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when <code>spec.os.name</code> is windows.

Property	Type	Description
<code>procMount</code>	<code>string</code>	<code>procMount</code> denotes the type of proc mount to use for the containers. The default value is <code>Default</code> which uses the container runtime defaults for readonly paths and masked paths. This requires the <code>ProcMountType</code> feature flag to be enabled. Note that this field cannot be set when <code>spec.os.name</code> is <code>windows</code>.
<code>readOnlyRootFilesystem</code>	<code>boolean</code>	Whether this container has a read-only root filesystem. Default is <code>false</code>. Note that this field cannot be set when <code>spec.os.name</code> is <code>windows</code>.
<code>runAsGroup</code>	<code>integer</code>	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in <code>PodSecurityContext</code>. If set in both <code>SecurityContext</code> and <code>PodSecurityContext</code>, the value specified in <code>SecurityContext</code> takes precedence. Note that this field cannot be set when <code>spec.os.name</code> is <code>windows</code>.
<code>runAsNonRoot</code>	<code>boolean</code>	Indicates that the container must run as a non-root user. If <code>true</code>, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or <code>false</code>, no such validation will be performed. May also be set in <code>PodSecurityContext</code>. If set in

Property	Type	Description
		both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seccompProfile	object	The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
<code>windowsOptions</code>	<code>object</code>	The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

`.spec.mysql.pxc.proxysql.containerSecurityContext.appArmorProfile`

Description

`appArmorProfile` is the AppArmor options to use by this container. If set, this profile overrides the pod's `appArmorProfile`. Note that this field cannot be set when `spec.os.name` is `windows`.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if <code>type</code> is "Localhost".

Property	Type	Description
<code>type</code>	<code>string</code>	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

`.spec.mysql.pxc.proxysql.containerSecurityContext.capabilities`

Description

The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when `spec.os.name` is windows.

Type

`object`

Property	Type	Description
<code>add</code>	<code>array</code>	Added capabilities
<code>drop</code>	<code>array</code>	Removed capabilities

`.spec.mysql.pxc.proxysql.containerSecurityContext.capabilities.add`

Description

Added capabilities

Type

array

.spec.mysql.pxc.proxysql.containerSecurityContext.capabilities.add[]**Description**

Capability represent POSIX capabilities type

Type

string

.spec.mysql.pxc.proxysql.containerSecurityContext.capabilities.drop**Description**

Removed capabilities

Type

array

.spec.mysql.pxc.proxysql.containerSecurityContext.capabilities.drop[]**Description**

Capability represent POSIX capabilities type

Type

string

.spec.mysql.pxc.proxysql.containerSecurityContext.seLinuxOptions**Description**

The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Type

object

Property	Type	Description
level	string	Level is SELinux level label that applies to the container.
role	string	Role is a SELinux role label that applies to the container.
type	string	Type is a SELinux type label that applies to the container.
user	string	User is a SELinux user label that applies to the container.

.spec.mysql.pxc.proxysql.containerSecurityContext.seccompProfile

Description

The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.

Type

object

Required

type

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.
type	string	type indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied.

.spec.mysql.pxc.proxysql.containerSecurityContext.windowsOptions

Description

The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

`.spec.mysql.pxc.proxysql.imagePullSecrets`

Type

array

`.spec.mysql.pxc.proxysql.imagePullSecrets[]`

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

`.spec.mysql.pxc.proxysql.labels`

Type

object

`.spec.mysql.pxc.proxysql.livenessProbes`

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	Exec specifies a command to execute in the
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Default value is 3. Minimum value is 1.
grpc	object	GRPC specifies a GRPC HealthCheckRequest to
httpGet	object	HTTPGet specifies an HTTP GET request to
initialDelaySeconds	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes
periodSeconds	integer	How often (in seconds) to perform the probe. Default is 10 seconds. Minimum value is 1.
successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Default is 1. Must be 1 for liveness and startup. Minimum value is 1.

Property	Type	Description
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies a connection to a TCP
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod needs gracefully upon probe failure. The grace period duration in seconds after the processes running pod are sent a termination signal and the time processes are forcibly halted with a kill signal. A value longer than the expected cleanup time process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. A value of zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. spec.terminationGracePeriodSeconds is used.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/lifecycle#container-probes

`.spec.mysql.pxc.proxysql.livenessProbes.exec`

Description

Exec specifies a command to execute in the container.

Type

`object`

Property	Type	Description
<code>command</code>	<code>array</code>	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.mysql.pxc.proxysql.livenessProbes.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

`.spec.mysql.pxc.proxysql.livenessProbes.exec.command[` `]`

Type

`string`

`.spec.mysql.pxc.proxysql.livenessProbes.grpc`

Description

GRPC specifies a GRPC HealthCheckRequest.

Type

`object`**Required**`port`

Property	Type	Description
<code>port</code>	<code>integer</code>	Port number of the gRPC service. Number must be in the range 1 to 65535.
<code>service</code>	<code>string</code>	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

.spec.mysql.pxc.proxysql.livenessProbes.httpGet

Description

HTTPGet specifies an HTTP GET request to perform.

Type`object`**Required**`port`

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.

Property	Type	Description
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.mysql.pxc.proxysql.livenessProbes.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.mysql.pxc.proxysql.livenessProbes.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.mysql.pxc.proxysql.livenessProbes.tcpSocket

Description

TCPSocket specifies a connection to a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

.spec.mysql.pxc.proxysql.loadBalancerSourceRanges

Type

array

.spec.mysql.pxc.proxysql.loadBalancerSourceRanges[]

Type

string

.spec.mysql.pxc.proxysql.nodeSelector

Type

object

.spec.mysql.pxc.proxysql.podDisruptionBudget

Type

object

Property	Type	Description
maxUnavailable		
minAvailable		

.spec.mysql.pxc.proxysql.podSecurityContext

Description

PodSecurityContext holds pod-level security attributes and common container settings. Some fields are also present in container.securityContext. Field values of container.securityContext take precedence over field values of PodSecurityContext.

Type

object

Property	Type	Description
appArmorProfile	object	appArmorProfile is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.
fsGroup	integer	A special supplemental group that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod: 1. The owning GID will be the FSGroup 2. The setgid bit is set (new files created in the volume will be owned by FSGroup) 3. The permission bits are OR'd with rw-rw-- -- If unset, the Kubelet will not modify the ownership and permissions of any volume. Note that this field cannot be set when spec.os.name is windows.
fsGroupChangePolicy	string	fsGroupChangePolicy defines behavior of changing ownership and permission of the volume before being exposed inside Pod. This field will only apply to volume types which support fsGroup based ownership(and permissions). It will have no effect on ephemeral volume types such as: secret, configmaps and emptydir. Valid values are "OnRootMismatch" and "Always". If not

Property	Type	Description
		specified, "Always" is used. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this

Property	Type	Description
		field cannot be set when spec.os.name is windows.
seLinuxChangePolicy	string	seLinuxChangePolicy defines how the container's SELinux label is applied to all volumes used by the Pod. It has no effect on nodes that do not support SELinux or to volumes does not support SELinux. Valid values are "MountOption" and "Recursive". "Recursive" means relabeling of all files on all Pod volumes by the container runtime. This may be slow for large volumes, but allows mixing privileged and unprivileged Pods sharing the same volume on the same node. "MountOption" mounts all eligible Pod volumes with <code>-o context</code> mount option. This requires all Pods that share the same volume to use the same SELinux label. It is not possible to share the same volume among privileged and unprivileged Pods. Eligible volumes are in-tree FibreChannel and iSCSI volumes, and all CSI volumes whose CSI driver announces SELinux support by setting spec.seLinuxMount: true in their CSIDriver instance. Other volumes are always re-labelled recursively. "MountOption" value is allowed only when SELinuxMount feature gate is enabled. If not specified and SELinuxMount feature gate is enabled, "MountOption" is used. If not specified and SELinuxMount feature

Property	Type	Description
		gate is disabled, "MountOption" is used for ReadWriteOncePod volumes and "Recursive" for all other volumes. This field affects only Pods that have SELinux label set, either in PodSecurityContext or in SecurityContext of all containers. All Pods that use the same volume should use the same seLinuxChangePolicy, otherwise some pods can get stuck in ContainerCreating state. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
seccompProfile	object	The seccomp options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
<code>supplementalGroups</code>	<code>array</code>	A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.
<code>supplementalGroupsPolicy</code>	<code>string</code>	Defines how supplemental groups of the first container processes are calculated. Valid values are "Merge" and "Strict". If not specified, "Merge" is used. (Alpha) Using the field requires the SupplementalGroupsPolicy feature gate to be enabled and the container runtime must implement support for this feature. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
<code>sysctls</code>	<code>array</code>	Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.
<code>windowsOptions</code>	<code>object</code>	The Windows specific settings applied to all containers. If unspecified, the options within a container's SecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

`.spec.mysql.pxc.proxysql.podSecurityContext.appArmorProfile`

Description

appArmorProfile is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
<code>type</code>	<code>string</code>	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

`.spec.mysql.pxc.proxysql.podSecurityContext.seLinuxOptions`

Description

The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Property	Type	Description
<code>level</code>	<code>string</code>	Level is SELinux level label that applies to the container.

Property	Type	Description
<code>role</code>	<code>string</code>	Role is a SELinux role label that applies to the container.
<code>type</code>	<code>string</code>	Type is a SELinux type label that applies to the container.
<code>user</code>	<code>string</code>	User is a SELinux user label that applies to the container.

`.spec.mysql.pxc.proxysql.podSecurityContext.seccompProfile`

Description

The seccomp options to use by the containers in this pod. Note that this field cannot be set when `spec.os.name` is `windows`.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.

Property	Type	Description
<code>type</code>	<code>string</code>	type indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied.

`.spec.mysql.pxc.proxysql.podSecurityContext.supplementalGroups`

Description

A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.

Type

`array`

`.spec.mysql.pxc.proxysql.podSecurityContext.supplementalGroups[]`

Type

`integer`

`.spec.mysql.pxc.proxysql.podSecurityContext.sysctls`

Description

Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.

Type

array

.spec.mysql.pxc.proxysql.podSecurityContext.sysctls[]

Description

Sysctl defines a kernel parameter to be set

Type

object

Required

name value

Property	Type	Description
name	string	Name of a property to set
value	string	Value of a property to set

.spec.mysql.pxc.proxysql.podSecurityContext.windowsOptions

Description

The Windows specific settings applied to all containers. If unspecified, the options within a container's SecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa [↗]) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.

Property	Type	Description
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entryptoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

`.spec.mysql.pxc.proxysql.readinessProbes`

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

`object`

Property	Type	Description
<code>exec</code>	<code>object</code>	Exec specifies a command to execute in the
<code>failureThreshold</code>	<code>integer</code>	Minimum consecutive failures for the probe to be considered failed after having succeeded. Default value is 3. Minimum value is 1.
<code>grpc</code>	<code>object</code>	GRPC specifies a GRPC HealthCheckRequest to send to the service.

Property	Type	Description
<code>httpGet</code>	<code>object</code>	HTTPGet specifies an HTTP GET request to
<code>initialDelaySeconds</code>	<code>integer</code>	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/lifecycle#container-probes
<code>periodSeconds</code>	<code>integer</code>	How often (in seconds) to perform the probe. Default is 10 seconds. Minimum value is 1.
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe to be considered successful after having failed. Default is 1. Must be 1 for liveness and startup. Minimum value is 1.
<code>tcpSocket</code>	<code>object</code>	TCP Socket specifies a connection to a TCP port.
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod needs to gracefully shutdown before it is killed. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. It specifies that the pod needs to terminate gracefully within this time by default. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via the kill signal.

Property	Type	Description
		(no opportunity to shut down). This is a beta requires enabling ProbeTerminationGracePeriodSeconds. Minimum value is 1. spec.terminationGracePeriodSeconds is use
timeoutSeconds	integer	Number of seconds after which the probe timeout. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/lifecycle#container-probes

.spec.mysql.pxc.proxysql.readinessProbes.exec

Description

Exec specifies a command to execute in the container.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.mysql.pxc.proxysql.readinessProbes.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.mysql.pxc.proxysql.readinessProbes.exec.command[]`

Type

string

`.spec.mysql.pxc.proxysql.readinessProbes.grpc`

Description

GRPC specifies a GRPC HealthCheckRequest.

Type

object

Required

port

Property	Type	Description
<code>port</code>	<code>integer</code>	Port number of the gRPC service. Number must be in the range 1 to 65535.
<code>service</code>	<code>string</code>	Service is the name of the service to place in the gRPC HealthCheckRequest (see

Property	Type	Description
		https://github.com/grpc/grpc/blob/master/doc/health-checking.md ). If this is not specified, the default behavior is defined by gRPC.

.spec.mysql.pxc.proxysql.readinessProbes.httpGet

Description

HTTPGet specifies an HTTP GET request to perform.

Type

object

Required

port

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
path	string	Path to access on the HTTP server.
port		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

Property	Type	Description
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.mysql.pxc.proxysql.readinessProbes.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.mysql.pxc.proxysql.readinessProbes.httpGet.httpHeaders[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name`

`value`

Property	Type	Description
<code>name</code>	<code>string</code>	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
<code>value</code>	<code>string</code>	The header field value

`.spec.mysql.pxc.proxysql.readinessProbes.tcpSocket`

Description

TCPSocket specifies a connection to a TCP port.

Type

object

Required

port

Property	Type	Description
<code>host</code>	<code>string</code>	Optional: Host name to connect to, defaults to the pod IP.
<code>port</code>		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.mysql.pxc.proxysql.resources`

Type

object

Property	Type	Description
<code>limits</code>	<code>object</code>	
<code>requests</code>	<code>object</code>	

`.spec.mysql.pxc.proxysql.resources.limits`

Type

`object`

Property	Type	Description
<code>cpu</code>	<code>string</code>	
<code>ephemeral-storage</code>	<code>string</code>	
<code>memory</code>	<code>string</code>	

`.spec.mysql.pxc.proxysql.resources.requests`

Type

`object`

Property	Type	Description
<code>cpu</code>	<code>string</code>	
<code>ephemeral-storage</code>	<code>string</code>	
<code>memory</code>	<code>string</code>	

`.spec.mysql.pxc.proxysql.serviceAnnotations`

Type

`object`

`.spec.mysql.pxc.proxysql.sidecarResources`

Type

`object`

Property	Type	Description
limits	object	
requests	object	

.spec.mysql.pxc.proxysql.sidecarResources.limits

Type

object

Property	Type	Description
cpu	string	
ephemeral-storage	string	
memory	string	

.spec.mysql.pxc.proxysql.sidecarResources.requests

Type

object

Property	Type	Description
cpu	string	
ephemeral-storage	string	
memory	string	

.spec.mysql.pxc.proxysql.startupProbe

Type

`object`

Property	Type	Description
<code>failureThreshold</code>	<code>integer</code>	
<code>initialDelaySeconds</code>	<code>integer</code>	
<code>periodSeconds</code>	<code>integer</code>	

`.spec.mysql.pxc.proxysql.tolerations`

Type

`array`

`.spec.mysql.pxc.proxysql.tolerations[]`

Description

The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.

Type

`object`

Property	Type	Description
<code>effect</code>	<code>string</code>	Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.
<code>key</code>	<code>string</code>	Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is

Property	Type	Description
		empty, operator must be Exists; this combination means to match all values and all keys.
<code>operator</code>	<code>string</code>	Operator represents a key's relationship to the value. Valid operators are Exists and Equal. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category.
<code>tolerationSeconds</code>	<code>integer</code>	TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is not set, which means tolerate the taint forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.
<code>value</code>	<code>string</code>	Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

`.spec.mysql.pxc.proxysql.volumeSpec`

Type

`object`

Property	Type	Description
<code>emptyDir</code>	<code>object</code>	EmptyDir to use as data volume for mysql. EmptyDir represents a temporary directory that

Property	Type	Description
		shares a pod's lifetime.
<code>hostPath</code>	<code>object</code>	HostPath to use as data volume for mysql. HostPath represents a pre-existing file or directory on the host machine that is directly exposed to the container.
<code>persistentVolumeClaim</code>	<code>object</code>	PersistentVolumeClaim to specify PVC spec for the volume for mysql data. It has the highest level of precedence, followed by HostPath and EmptyDir. And represents the PVC specification.

`.spec.mysql.pxc.proxysql.volumeSpec.emptyDir`

Description

EmptyDir to use as data volume for mysql. EmptyDir represents a temporary directory that shares a pod's lifetime.

Type

`object`

Property	Type	Description
<code>medium</code>	<code>string</code>	medium represents what type of storage medium should back this directory. The default is "" which means to use the node's default medium. Must be an empty string (default) or Memory. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir

Property	Type	Description
<code>sizeLimit</code>		sizeLimit is the total amount of local storage required for this EmptyDir volume. The size limit is also applicable for memory medium. The maximum usage on memory medium EmptyDir would be the minimum value between the SizeLimit specified here and the sum of memory limits of all containers in a pod. The default is nil which means that the limit is undefined. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir

.spec.mysql.pxc.proxysql.volumeSpec.hostPath

Description

HostPath to use as data volume for mysql. HostPath represents a pre-existing file or directory on the host machine that is directly exposed to the container.

Type

`object`

Required

`path`

Property	Type	Description
<code>path</code>	<code>string</code>	path of the directory on the host. If the path is a symlink, it will follow the link to the real path. More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath
<code>type</code>	<code>string</code>	type for HostPath Volume Defaults to "" More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath

.spec.mysql.pxc.proxysql.volumeSpec.persistentVolumeClaim

Description

PersistentVolumeClaim to specify PVC spec for the volume for mysql data. It has the highest level of precedence, followed by HostPath and EmptyDir. And represents the PVC specification.

Type

object

Property	Type	Description
<code>accessModes</code>	array	accessModes contains the desired access modes volume should have. More info: https://kubernetes.io/docs/concepts/storage/persistentvolumes#access-modes-1
<code>dataSource</code>	object	dataSource field can be used to specify either: • An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) • An existing PVC (PersistentVolumeClaim) If the provisioner or an external controller can support the specified data source, it will create a new volume based on the contents of the specified data source. When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If the namespace is specified, then dataSourceRef will not be copied to dataSource.

Property	Type	Description
dataSourceRef	object	dataSourceRef specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a PersistentVolumeClaim object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will require the functionality of the dataSource field and as such both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in dataSourceRef, dataSource isn't set to the same value and must be empty. There are three important differences between dataSource and dataSourceRef • While dataSource only allows two specific type objects, dataSourceRef allows any non-core object as well as PersistentVolumeClaim objects. • While dataSource ignores disallowed values (dropping them), dataSourceRef preserves all values, and generates an error if a disallowed value is specified. • While dataSource only allows local objects, dataSourceRef allows objects in any namespace (Beta) Using this field requires the AnyVolumeDataSource feature gate to be enabled (Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

Property	Type	Description
resources	object	resources represents the minimum resources the volume should have. If RecoverVolumeExpansionFailure feature is enable users are allowed to specify resource requirement that are lower than previous value but must still be higher than capacity recorded in the status field of claim. More info: https://kubernetes.io/docs/concepts/storage/persistentvolumes#resources
selector	object	selector is a label query over volumes to consider binding.
storageClassName	string	storageClassName is the name of the StorageClass required by the claim. More info: https://kubernetes.io/docs/concepts/storage/persistentvolumes#class-1
volumeAttributesClassName	string	volumeAttributesClassName may be used to set the VolumeAttributesClass used by this claim. If specified the CSI driver will create or update the volume with attributes defined in the corresponding VolumeAttributesClass. This has a different purpose than storageClassName, it can be changed after the claim is created. An empty string value means that VolumeAttributesClass will be applied to the claim it's not allowed to reset this field to empty string once it is set. If unspecified and the PersistentVolumeClaim is unbound, the default VolumeAttributesClass will be

Property	Type	Description
		by the persistentvolume controller if it exists. If the resource referred to by volumeAttributesClass does not exist, this PersistentVolumeClaim will be set to Pending state, as reflected by the modifyVolumeStatus field, until such as a resource exists. More info: https://kubernetes.io/docs/concepts/storage/volume-attributes-classes/ (Beta) Using this field requires the VolumeAttributesClass feature gate to be enabled (off by default).
<code>volumeMode</code>	<code>string</code>	volumeMode defines what type of volume is required by the claim. Value of Filesystem is implied when not included in claim spec.
<code>volumeName</code>	<code>string</code>	volumeName is the binding reference to the PersistentVolume backing this claim.

`.spec.mysql.pxc.proxysql.volumeSpec.persistentVolumeClaim.accessModes`

Description

accessModes contains the desired access modes the volume should have. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1>

Type

`array`

`.spec.mysql.pxc.proxysql.volumeSpec.persistentVolumeClaim.accessModes[]`

Type

string

.spec.mysql.pxc.proxysql.volumeSpec.persistentVolumeClaim.dataSource

Description

dataSource field can be used to specify either: * An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) * An existing PVC (PersistentVolumeClaim) If the provisioner or an external controller can support the specified data source, it will create a new volume based on the contents of the specified data source. When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If the namespace is specified, then dataSourceRef will not be copied to dataSource.

Type

object

Required

kind

name

Property	Type	Description
apiGroup	string	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
kind	string	Kind is the type of resource being referenced
name	string	Name is the name of resource being referenced

.spec.mysql.pxc.proxysql.volumeSpec.persistentVolumeClaim.dataSourceRef

Description

dataSourceRef specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a PersistentVolumeClaim object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replace the functionality of the dataSource field and as such if both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in dataSourceRef, dataSource isn't set to the same value and must be empty. There are three important differences between dataSource and dataSourceRef: * While dataSource only allows two specific types of objects, dataSourceRef allows any non-core object, as well as PersistentVolumeClaim objects. * While dataSource ignores disallowed values (dropping them), dataSourceRef preserves all values, and generates an error if a disallowed value is specified. * While dataSource only allows local objects, dataSourceRef allows objects in any namespaces. (Beta) Using this field requires the AnyVolumeDataSource feature gate to be enabled. (Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

Type

object

Required

kind

name

Property	Type	Description
apiGroup	string	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.

Property	Type	Description
<code>kind</code>	<code>string</code>	Kind is the type of resource being referenced
<code>name</code>	<code>string</code>	Name is the name of resource being referenced
<code>namespace</code>	<code>string</code>	Namespace is the namespace of resource being referenced Note that when a namespace is specified, a <code>gateway.networking.k8s.io/ReferenceGrant</code> object is required in the referent namespace to allow that namespace's owner to accept the reference. See the <code>ReferenceGrant</code> documentation for details. (Alpha) This field requires the <code>CrossNamespaceVolumeDataSource</code> feature gate to be enabled.

`.spec.mysql.pxc.proxysql.volumeSpec.persistentVolumeClaim.resources`

Description

`resources` represents the minimum resources the volume should have. If `RecoverVolumeExpansionFailure` feature is enabled users are allowed to specify resource requirements that are lower than previous value but must still be higher than capacity recorded in the status field of the claim. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources>

Type

`object`

Property	Type	Description
<code>limits</code>	<code>object</code>	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗
<code>requests</code>	<code>object</code>	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

`.spec.mysql.pxc.proxysql.volumeSpec.persistentVolumeClaim.resources.limits`

Description

Limits describes the maximum amount of compute resources allowed. More info:
<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.mysql.pxc.proxysql.volumeSpec.persistentVolumeClaim.resources.requests`

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an

implementation-defined value. Requests cannot exceed Limits. More info:
<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

`.spec.mysql.pxc.proxysql.volumeSpec.persistentVolumeClaim.selector`

Description

selector is a label query over volumes to consider for binding.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.pxc.proxysql.volumeSpec.persistentVolumeClaim.selector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.mysql.pxc.proxysql.volumeSpec.persistentVolumeClaim.selector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.mysql.pxc.proxysql.volumeSpec.persistentVolumeClaim.selector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.pxc.proxysql.volumeSpec.persistentVolumeClaim.selector.matchExpressions[].values[]

Type

string

.spec.mysql.pxc.proxysql.volumeSpec.persistentVolumeClaim.selector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mysql.pxc.pxc

Type

object

Property	Type	Description
affinity	object	
annotations	object	

Property	Type	Description
autoRecovery	boolean	
configuration	string	
containerSecurityContext	object	SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.
enableProbes	boolean	
enabled	boolean	
externalTrafficPolicy	string	for backwards compat
forceUnsafeBootstrap	boolean	
gracePeriod	integer	
image	string	
imagePullPolicy	string	PullPolicy describes a policy for if/when to pull a container image
imagePullSecrets	array	
labels	object	
livenessDelaySec	integer	

Property	Type	Description
<code>livenessProbes</code>	<code>object</code>	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
<code>loadBalancerSourceRanges</code>	<code>array</code>	
<code>nodeSelector</code>	<code>object</code>	
<code>podDisruptionBudget</code>	<code>object</code>	
<code>podSecurityContext</code>	<code>object</code>	PodSecurityContext holds pod-level security attributes and common container settings. Some fields are also present in container.securityContext. Field values of container.securityContext take precedence over field values of PodSecurityContext.
<code>priorityClassName</code>	<code>string</code>	
<code>readinessDelaySec</code>	<code>integer</code>	
<code>readinessProbes</code>	<code>object</code>	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
<code>replicasExternalTrafficPolicy</code>	<code>string</code>	for backwards compat

Property	Type	Description
<code>replicasServiceType</code>	<code>string</code>	Service Type string describes ingress methods for a service
<code>replicationChannels</code>	<code>array</code>	
<code>resources</code>	<code>object</code>	
<code>schedulerName</code>	<code>string</code>	
<code>serverMaxConnections</code>	<code>integer</code>	
<code>serviceAccountName</code>	<code>string</code>	
<code>serviceAnnotations</code>	<code>object</code>	
<code>serviceType</code>	<code>string</code>	Service Type string describes ingress methods for a service
<code>sidecarResources</code>	<code>object</code>	
<code>size</code>	<code>integer</code>	
<code>sslInternalSecretName</code>	<code>string</code>	
<code>sslSecretName</code>	<code>string</code>	
<code>startupProbe</code>	<code>object</code>	
<code>tolerations</code>	<code>array</code>	
<code>vaultSecretName</code>	<code>string</code>	
<code>volumeSpec</code>	<code>object</code>	
<code>writeNode</code>	<code>string</code>	

.spec.mysql.pxc.pxc.affinity

Type

object

Property	Type	Description
advanced	object	Affinity is a group of affinity scheduling rules.
antiAffinityTopologyKey	string	

.spec.mysql.pxc.pxc.affinity.advanced

Description

Affinity is a group of affinity scheduling rules.

Type

object

Property	Type	Description
nodeAffinity	object	Describes node affinity scheduling rules for the pod.
podAffinity	object	Describes pod affinity scheduling rules (e.g. co-locate this pod in the same node, zone, etc. as some other pod(s)).
podAntiAffinity	object	Describes pod anti-affinity scheduling rules (e.g. avoid putting this pod in the same node, zone, etc. as some other pod(s)).

.spec.mysql.pxc.pxc.affinity.advanced.nodeAffinity

Description

Describes node affinity scheduling rules for the pod.

Type

object

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution affinity expressions, etc.) compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest

Property	Type	Description
		sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>object</code>	If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to an update), the system may or may not try to eventually evict the pod from its node.

`.spec.mysql.pxc.pxc.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, `requiredDuringScheduling` affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding `matchExpressions`; the node(s) with the highest sum are the most preferred.

Type

`array`

.spec.mysql.pxc.pxc.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

Description

An empty preferred scheduling term matches all objects with implicit weight 0 (i.e. it's a no-op). A null preferred scheduling term matches no objects (i.e. is also a no-op).

Type

object

Required

preference weight

Property	Type	Description
preference	object	A node selector term, associated with the corresponding weight.
weight	integer	Weight associated with matching the corresponding nodeSelectorTerm, in the range 1-100.

.spec.mysql.pxc.pxc.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference

Description

A node selector term, associated with the corresponding weight.

Type

object

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	A list of node selector requirements by node's labels.
<code>matchFields</code>	<code>array</code>	A list of node selector requirements by node's fields.

`.spec.mysql.pxc.pxc.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions`

Description

A list of node selector requirements by node's labels.

Type

`array`

`.spec.mysql.pxc.pxc.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[]`

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

`key`

`operator`

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist, Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.mysql.pxc.pxc.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.pxc.pxc.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.

matchExpressions[].values[]

Type

string

.spec.mysql.pxc.pxc.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields

Description

A list of node selector requirements by node's fields.

Type

array

.spec.mysql.pxc.pxc.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[]

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	The label key that the selector applies to.

Property	Type	Description
<code>operator</code>	<code>string</code>	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
<code>values</code>	<code>array</code>	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

`.spec.mysql.pxc.pxc.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values`

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.mysql.pxc.pxc.affinity.advanced.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values[]`

Type

`string`

.spec.mysql.pxc.pxc.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to an update), the system may or may not try to eventually evict the pod from its node.

Type

object

Required

nodeSelectorTerms

Property	Type	Description
nodeSelectorTerms	array	Required. A list of node selector terms. The terms are ORed.

.spec.mysql.pxc.pxc.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms

Description

Required. A list of node selector terms. The terms are ORed.

Type

array

.spec.mysql.pxc.pxc.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms

rTerms[]

Description

A null or empty node selector term matches no objects. The requirements of them are ANDed. The TopologySelectorTerm type implements a subset of the NodeSelectorTerm.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	A list of node selector requirements by node's labels.
<code>matchFields</code>	array	A list of node selector requirements by node's fields.

`.spec.mysql.pxc.pxc.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions`

Description

A list of node selector requirements by node's labels.

Type

array

`.spec.mysql.pxc.pxc.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[]`

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.mysql.pxc.pxc.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.pxc.pxc.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values[]

Type

string

.spec.mysql.pxc.pxc.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields

Description

A list of node selector requirements by node's fields.

Type

array

.spec.mysql.pxc.pxc.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[]

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
values	array	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

.spec.mysql.pxc.pxc.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values

Description

An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.pxc.pxc.affinity.advanced.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values

rTerms[].matchFields[].values[]

Type

string

.spec.mysql.pxc.pxc.affinity.advanced.podAffinity

Description

Describes pod affinity scheduling rules (e.g. co-locate this pod in the same node, zone, etc. as some other pod(s)).

Type

object

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution affinity expressions, etc.), compute a sum by iterating through the elements of this field and

Property	Type	Description
		adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	If the affinity requirement: specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirement: specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

`.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

`array`

`.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Type

`object`

Required

`podAffinityTerm``weight`

Property	Type	Description
<code>podAffinityTerm</code>	<code>object</code>	Required. A pod affinity term, associated with the corresponding weight.

Property	Type	Description
<code>weight</code>	<code>integer</code>	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

`.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm`

Description

Required. A pod affinity term, associated with the corresponding weight.

Type

`object`

Required

`topologyKey`

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be

Property	Type	Description
		ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Property	Type	Description
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityT

erm.labelSelector.matchExpressions[].values[]

Type

string

.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]

Type

string

.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]

Type

string

`.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]

`erm.namespaceSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

`.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]`

Type

string

`.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

`.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces`

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

string

.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Type

array

.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

Defines a set of pods (namely those matching the `labelSelector` relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key `<topologyKey>` matches that of any node on which a pod of the set of pods is running

Type

object

Required

topologyKey

Property	Type	Description
<code>labelSelector</code>	object	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	array	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>mismatchLabelKeys</code>	array	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The

Property	Type	Description
		keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both <code>mismatchLabelKeys</code> and <code>labelSelector</code>. Also, <code>mismatchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. This is a beta field and requires enabling <code>MatchLabelKeysInPodAffinity</code> feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the <code>namespaces</code> field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector (<code>{}</code>) matches all namespaces.
<code>namespaces</code>	<code>array</code>	<code>namespaces</code> specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by <code>namespaceSelector</code>. null or empty namespaces list and null <code>namespaceSelector</code> means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the <code>labelSelector</code> in the specified namespaces, where co-

Property	Type	Description
		located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector`

r.matchExpressions

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.

Property	Type	Description
<code>values</code>	<code>array</code>	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]`

Type

`string`

`.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.require dDuringSchedulingIgnoredDuringExecution[].matchLabel Keys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.require dDuringSchedulingIgnoredDuringExecution[].matchLabel Keys[]

Type

string

.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.require dDuringSchedulingIgnoredDuringExecution[].mismatchLa

belKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.require dDuringSchedulingIgnoredDuringExecution[].mismatchLa belKeys[]

Type

string

.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.require dDuringSchedulingIgnoredDuringExecution[].namespace Selector

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.require dDuringSchedulingIgnoredDuringExecution[].namespace Selector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.require dDuringSchedulingIgnoredDuringExecution[].namespace Selector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespace

Selector.matchExpressions[].values[]

Type

string

**.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.require
dDuringSchedulingIgnoredDuringExecution[].namespace
Selector.matchLabels**

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

**.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.require
dDuringSchedulingIgnoredDuringExecution[].namespace
S**

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

**.spec.mysql.pxc.pxc.affinity.advanced.podAffinity.require
dDuringSchedulingIgnoredDuringExecution[].namespace
s[]**

Type

string

.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity

Description

Describes pod anti-affinity scheduling rules (e.g. avoid putting this pod in the same node, zone, etc. as some other pod(s)).

Type

object

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution, anti-affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods

Property	Type	Description
		which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling anti-affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

array

.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

Description

The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Type

object

Required

podAffinityTerm

weight

Property	Type	Description
podAffinityTerm	object	Required. A pod affinity term, associated with the corresponding weight.
weight	integer	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm

Description

Required. A pod affinity term, associated with the corresponding weight.

Type

object

Required

topologyKey

Property	Type	Description
labelSelector	object	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
matchLabelKeys	array	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key in (value) to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Property	Type	Description
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Property	Type	Description
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the <code>labelSelector</code> in the specified namespaces, where co-located is defined as running on a node whose value of the label with key <code>topologyKey</code> matches that of any node on which any of the selected pods is running. Empty <code>topologyKey</code> is not allowed.

`.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector`

Description

A label query over a set of resources, in this case pods. If it's null, this `PodAffinityTerm` matches with no Pods.

Type

`object`

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.

Property	Type	Description
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]

Type

string

.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]

Type

string

.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffi

Term.mismatchLabelKeys

Description

MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Type

array

.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]

Type

string

.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
<code>matchExpressions</code>	<code>array</code>	<code>matchExpressions</code> is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	<code>object</code>	<code>matchLabels</code> is a map of {key,value} pairs. A single {key,value} in the <code>matchLabels</code> map is equivalent to an element of <code>matchExpressions</code> , whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions`

Description

`matchExpressions` is a list of label selector requirements. The requirements are ANDed.

Type

`array`

`.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

`object`

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffi

affinityTerm.namespaceSelector.matchExpressions[].values[]

Type

string

.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

string

.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Type

array

.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running

Type

object

Required

topologyKey

Property	Type	Description
<code>labelSelector</code>	<code>object</code>	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
<code>matchLabelKeys</code>	<code>array</code>	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key in (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>mismatchLabelKeys</code>	<code>array</code>	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with <code>labelSelector</code> as <code>key notin (value)</code> to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field

Property	Type	Description
		and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
<code>namespaceSelector</code>	<code>object</code>	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
<code>namespaces</code>	<code>array</code>	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSel`

ector

Description

A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

`.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSel`

ector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This

array is replaced during a strategic merge patch.

Type

array

.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]

Type

string

.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti)

affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both `matchLabelKeys` and `labelSelector`. Also, `matchLabelKeys` cannot be set when `labelSelector` isn't set. This is a beta field and requires enabling `MatchLabelKeysInPodAffinity` feature gate (enabled by default).

Type

array

`.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]`

Type

string

`.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys`

Description

`MismatchLabelKeys` is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both `mismatchLabelKeys` and `labelSelector`. Also, `mismatchLabelKeys` cannot be set when `labelSelector` isn't set. This is a beta field and requires enabling `MatchLabelKeysInPodAffinity` feature gate (enabled by default).

Type

array

`.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys`

hLabelKeys[]

Type

string

.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

Property	Type	Description
matchExpressions	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
matchLabels	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namesp

aceSelector.matchExpressions

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.

Property	Type	Description
<code>values</code>	<code>array</code>	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

`array`

`.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]`

Type

`string`

`.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels`

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces

Description

namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Type

array

.spec.mysql.pxc.pxc.affinity.advanced.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]

Type

string

.spec.mysql.pxc.pxc.annotations

Type

object

.spec.mysql.pxc.pxc.containerSecurityContext

Description

SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.

Type

object

Property	Type	Description
<code>allowPrivilegeEscalation</code>	<code>boolean</code>	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the <code>no_new_privs</code> flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: 1. run as Privileged 2. has CAP_SYS_ADMIN Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>appArmorProfile</code>	<code>object</code>	<code>appArmorProfile</code> is the AppArmor options to use by this container. If set, this profile overrides the pod's <code>appArmorProfile</code>. Note that this field cannot be set when <code>spec.os.name</code> is windows.
<code>capabilities</code>	<code>object</code>	The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when <code>spec.os.name</code> is windows.

Property	Type	Description
<code>privileged</code>	<code>boolean</code>	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.
<code>procMount</code>	<code>string</code>	procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows.
<code>readOnlyRootFilesystem</code>	<code>boolean</code>	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
<code>runAsGroup</code>	<code>integer</code>	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
<code>runAsNonRoot</code>	<code>boolean</code>	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
<code>runAsUser</code>	<code>integer</code>	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
<code>seLinuxOptions</code>	<code>object</code>	The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
<code>seccompProfile</code>	<code>object</code>	The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.
<code>windowsOptions</code>	<code>object</code>	The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

`.spec.mysql.pxc.pxc.containerSecurityContext.appArmorProfile`

Description

`appArmorProfile` is the AppArmor options to use by this container. If set, this profile overrides the pod's `appArmorProfile`. Note that this field cannot be set when `spec.os.name` is windows.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
<code>type</code>	<code>string</code>	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

`.spec.mysql.pxc.pxc.containerSecurityContext.capabilitie` `S`

Description

The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when `spec.os.name` is windows.

Type

`object`

Property	Type	Description
<code>add</code>	<code>array</code>	Added capabilities
<code>drop</code>	<code>array</code>	Removed capabilities

.spec.mysql.pxc.pxc.containerSecurityContext.capabilities.add

Description

Added capabilities

Type

array

.spec.mysql.pxc.pxc.containerSecurityContext.capabilities.add[]

Description

Capability represent POSIX capabilities type

Type

string

.spec.mysql.pxc.pxc.containerSecurityContext.capabilities.drop

Description

Removed capabilities

Type

array

.spec.mysql.pxc.pxc.containerSecurityContext.capabilities.drop[]

Description

Capability represent POSIX capabilities type

Type

string

.spec.mysql.pxc.pxc.containerSecurityContext.seLinuxOptions

Description

The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Type

object

Property	Type	Description
level	string	Level is SELinux level label that applies to the container.
role	string	Role is a SELinux role label that applies to the container.
type	string	Type is a SELinux type label that applies to the container.
user	string	User is a SELinux user label that applies to the container.

.spec.mysql.pxc.pxc.containerSecurityContext.seccompProfile

Description

The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.

Type

object

Required

type

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.
type	string	type indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied.

.spec.mysql.pxc.pxc.containerSecurityContext.windowsOptions

Description

The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the

value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the

Property	Type	Description
		value specified in SecurityContext takes precedence.

`.spec.mysql.pxc.pxc.imagePullSecrets`

Type

array

`.spec.mysql.pxc.pxc.imagePullSecrets[]`

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

`.spec.mysql.pxc.pxc.labels`

Type

object

.spec.mysql.pxc.pxc.livenessProbes

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
<code>exec</code>	object	Exec specifies a command to execute in the
<code>failureThreshold</code>	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Default value is 3. Minimum value is 1.
<code>grpc</code>	object	GRPC specifies a GRPC HealthCheckRequest to
<code>httpGet</code>	object	HTTPGet specifies an HTTP GET request to
<code>initialDelaySeconds</code>	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/controllers/lifecycle#container-probes
<code>periodSeconds</code>	integer	How often (in seconds) to perform the probe. Default is 10 seconds. Minimum value is 1.

Property	Type	Description
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies a connection to a TCP endpoint.
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod needs to gracefully shutdown before it is terminated. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. A value greater than 0 indicates the grace period should be used. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. Defaults to 30 seconds. spec.terminationGracePeriodSeconds is used if specified.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More information: https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes

.spec.mysql.pxc.pxc.livenessProbes.exec

Description

Exec specifies a command to execute in the container.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.mysql.pxc.pxc.livenessProbes.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

.spec.mysql.pxc.pxc.livenessProbes.exec.command[]

Type

string

.spec.mysql.pxc.pxc.livenessProbes.grpc

Description

GRPC specifies a GRPC HealthCheckRequest.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

.spec.mysql.pxc.pxc.livenessProbes.httpGet

Description

HTTPGet specifies an HTTP GET request to perform.

Type

object

Required

port

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.mysql.pxc.pxc.livenessProbes.httpGet.httpHeaders`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.mysql.pxc.pxc.livenessProbes.httpGet.httpHeaders[ ]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.mysql.pxc.pxc.livenessProbes.tcpSocket

Description

TCPSocket specifies a connection to a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.mysql.pxc.pxc.loadBalancerSourceRanges`

Type

`array`

`.spec.mysql.pxc.pxc.loadBalancerSourceRanges[]`

Type

`string`

`.spec.mysql.pxc.pxc.nodeSelector`

Type

`object`

`.spec.mysql.pxc.pxc.podDisruptionBudget`

Type

`object`

Property	Type	Description
<code>maxUnavailable</code>		
<code>minAvailable</code>		

.spec.mysql.pxc.pxc.podSecurityContext

Description

PodSecurityContext holds pod-level security attributes and common container settings. Some fields are also present in container.securityContext. Field values of container.securityContext take precedence over field values of PodSecurityContext.

Type

object

Property	Type	Description
appArmorProfile	object	appArmorProfile is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.
fsGroup	integer	A special supplemental group that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod: 1. The owning GID will be the FSGroup 2. The setgid bit is set (new files created in the volume will be owned by FSGroup) 3. The permission bits are OR'd with rw-rw-- -- If unset, the Kubelet will not modify the ownership and permissions of any volume. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
<code>fsGroupChangePolicy</code>	<code>string</code>	<code>fsGroupChangePolicy</code> defines behavior of changing ownership and permission of the volume before being exposed inside Pod. This field will only apply to volume types which support <code>fsGroup</code> based ownership(and permissions). It will have no effect on ephemeral volume types such as: <code>secret</code>, <code>configmaps</code> and <code>emptydir</code>. Valid values are "OnRootMismatch" and "Always". If not specified, "Always" is used. Note that this field cannot be set when <code>spec.os.name</code> is <code>windows</code>.
<code>runAsGroup</code>	<code>integer</code>	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in <code>SecurityContext</code>. If set in both <code>SecurityContext</code> and <code>PodSecurityContext</code>, the value specified in <code>SecurityContext</code> takes precedence for that container. Note that this field cannot be set when <code>spec.os.name</code> is <code>windows</code>.
<code>runAsNonRoot</code>	<code>boolean</code>	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in <code>SecurityContext</code>. If set in both <code>SecurityContext</code> and <code>PodSecurityContext</code>,

Property	Type	Description
		the value specified in SecurityContext takes precedence.
<code>runAsUser</code>	<code>integer</code>	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
<code>seLinuxChangePolicy</code>	<code>string</code>	<code>seLinuxChangePolicy</code> defines how the container's SELinux label is applied to all volumes used by the Pod. It has no effect on nodes that do not support SELinux or to volumes does not support SELinux. Valid values are "MountOption" and "Recursive". "Recursive" means relabeling of all files on all Pod volumes by the container runtime. This may be slow for large volumes, but allows mixing privileged and unprivileged Pods sharing the same volume on the same node. "MountOption" mounts all eligible Pod volumes with <code>-o context</code> mount option. This requires all Pods that share the same volume to use the same SELinux label. It is not possible to share the same volume among privileged and unprivileged Pods.

Property	Type	Description
		Eligible volumes are in-tree FibreChannel and iSCSI volumes, and all CSI volumes whose CSI driver announces SELinux support by setting spec.selinuxMount: true in their CSIDriver instance. Other volumes are always re-labelled recursively. "MountOption" value is allowed only when SELinuxMount feature gate is enabled. If not specified and SELinuxMount feature gate is enabled, "MountOption" is used. If not specified and SELinuxMount feature gate is disabled, "MountOption" is used for ReadWriteOncePod volumes and "Recursive" for all other volumes. This field affects only Pods that have SELinux label set, either in PodSecurityContext or in SecurityContext of all containers. All Pods that use the same volume should use the same selinuxChangePolicy, otherwise some pods can get stuck in ContainerCreating state. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes

Property	Type	Description
		precedence for that container. Note that this field cannot be set when spec.os.name is windows.
seccompProfile	object	The seccomp options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.
supplementalGroups	array	A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.
supplementalGroupsPolicy	string	Defines how supplemental groups of the first container processes are calculated. Valid values are "Merge" and "Strict". If not specified, "Merge" is used. (Alpha) Using the field requires the SupplementalGroupsPolicy feature gate to be enabled and the container

Property	Type	Description
		runtime must implement support for this feature. Note that this field cannot be set when spec.os.name is windows.
<code>sysctls</code>	<code>array</code>	Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.
<code>windowsOptions</code>	<code>object</code>	The Windows specific settings applied to all containers. If unspecified, the options within a container's SecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

.spec.mysql.pxc.pxc.podSecurityContext.appArmorProfile

Description

appArmorProfile is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
<code>type</code>	<code>string</code>	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

`.spec.mysql.pxc.pxc.podSecurityContext.seLinuxOptions`

Description

The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Property	Type	Description
<code>level</code>	<code>string</code>	Level is SELinux level label that applies to the container.
<code>role</code>	<code>string</code>	Role is a SELinux role label that applies to the container.

Property	Type	Description
<code>type</code>	<code>string</code>	Type is a SELinux type label that applies to the container.
<code>user</code>	<code>string</code>	User is a SELinux user label that applies to the container.

`.spec.mysql.pxc.pxc.podSecurityContext.seccompProfile`

Description

The seccomp options to use by the containers in this pod. Note that this field cannot be set when `spec.os.name` is windows.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.
<code>type</code>	<code>string</code>	<code>type</code> indicates which kind of seccomp profile will be applied. Valid options are: Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default

Property	Type	Description
		profile should be used. Unconfined - no profile should be applied.

`.spec.mysql.pxc.pxc.podSecurityContext.supplementalGroups`

Description

A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.

Type

array

`.spec.mysql.pxc.pxc.podSecurityContext.supplementalGroups[]`

Type

integer

`.spec.mysql.pxc.pxc.podSecurityContext.sysctls`

Description

Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.

Type

array

`.spec.mysql.pxc.pxc.podSecurityContext.sysctls[]`

Description

Sysctl defines a kernel parameter to be set

Type

object

Required

name

value

Property	Type	Description
name	string	Name of a property to set
value	string	Value of a property to set

`.spec.mysql.pxc.pxc.podSecurityContext.windowsOptions`

Description

The Windows specific settings applied to all containers. If unspecified, the options within a container's SecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

.spec.mysql.pxc.pxc.readinessProbes

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
<code>exec</code>	object	Exec specifies a command to execute in the
<code>failureThreshold</code>	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Default value is 3. Minimum value is 1.
<code>grpc</code>	object	GRPC specifies a GRPC HealthCheckRequest to
<code>httpGet</code>	object	HTTPGet specifies an HTTP GET request to
<code>initialDelaySeconds</code>	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/controllers/lifecycle#container-probes
<code>periodSeconds</code>	integer	How often (in seconds) to perform the probe. Default is 10 seconds. Minimum value is 1.

Property	Type	Description
<code>successThreshold</code>	<code>integer</code>	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
<code>tcpSocket</code>	<code>object</code>	TCPSocket specifies a connection to a TCP endpoint.
<code>terminationGracePeriodSeconds</code>	<code>integer</code>	Optional duration in seconds the pod needs to gracefully shutdown before it is terminated. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. It specifies that the container managed liveness probes of the pod must not fail after this value longer than the expected cleanup time for the pod. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. Value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta feature and requires enabling ProbeTerminationGracePeriod. Minimum value is 1. Defaults to 30 seconds. spec.terminationGracePeriodSeconds is used if specified.
<code>timeoutSeconds</code>	<code>integer</code>	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/pods/lifecycle#container-probes

.spec.mysql.pxc.pxc.readinessProbes.exec

Description

Exec specifies a command to execute in the container.

Type

object

Property	Type	Description
command	array	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

`.spec.mysql.pxc.pxc.readinessProbes.exec.command`

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

array

`.spec.mysql.pxc.pxc.readinessProbes.exec.command[]`

Type

string

`.spec.mysql.pxc.pxc.readinessProbes.grpc`

Description

GRPC specifies a GRPC HealthCheckRequest.

Type

object

Required

port

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

.spec.mysql.pxc.pxc.readinessProbes.httpGet

Description

HTTPGet specifies an HTTP GET request to perform.

Type

object

Required

port

Property	Type	Description
<code>host</code>	<code>string</code>	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
<code>httpHeaders</code>	<code>array</code>	Custom headers to set in the request. HTTP allows repeated headers.
<code>path</code>	<code>string</code>	Path to access on the HTTP server.
<code>port</code>		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
<code>scheme</code>	<code>string</code>	Scheme to use for connecting to the host. Defaults to HTTP.

`.spec.mysql.pxc.pxc.readinessProbes.httpGet.httpHeaderS`

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

`.spec.mysql.pxc.pxc.readinessProbes.httpGet.httpHeaderS[]`

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

name

value

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

.spec.mysql.pxc.pxc.readinessProbes.tcpSocket

Description

TCPSocket specifies a connection to a TCP port.

Type

object

Required

port

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
<code>port</code>	<code>integer</code>	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

`.spec.mysql.pxc.pxc.replicationChannels`

Type

`array`

`.spec.mysql.pxc.pxc.replicationChannels[]`

Type

`object`

Property	Type	Description
<code>configuration</code>	<code>object</code>	
<code>enabled</code>	<code>boolean</code>	
<code>name</code>	<code>string</code>	
<code>sourceProxyList</code>	<code>array</code>	
<code>sourcesList</code>	<code>array</code>	

`.spec.mysql.pxc.pxc.replicationChannels[].configuration`

Type

`object`

Property	Type	Description
sourceConnectRetry	integer	
sourceRetryCount	integer	

.spec.mysql.pxc.pxc.replicationChannels[].sourceProxyList

Type

array

.spec.mysql.pxc.pxc.replicationChannels[].sourceProxyList[]

Type

object

Required

host hostName

Property	Type	Description
host	string	
hostName	string	
port	integer	
weight	integer	

.spec.mysql.pxc.pxc.replicationChannels[].sourcesList

Type

`array`

`.spec.mysql.pxc.pxc.replicationChannels[].sourcesList[]`

Type

`object`

Required

`host``hostName`

Property	Type	Description
<code>host</code>	<code>string</code>	
<code>hostName</code>	<code>string</code>	
<code>port</code>	<code>integer</code>	
<code>weight</code>	<code>integer</code>	

`.spec.mysql.pxc.pxc.resources`

Type

`object`

Property	Type	Description
<code>limits</code>	<code>object</code>	
<code>requests</code>	<code>object</code>	

`.spec.mysql.pxc.pxc.resources.limits`

Type

`object`

Property	Type	Description
cpu	string	
ephemeral-storage	string	
memory	string	

.spec.mysql.pxc.pxc.resources.requests

Type

object

Property	Type	Description
cpu	string	
ephemeral-storage	string	
memory	string	

.spec.mysql.pxc.pxc.serviceAnnotations

Type

object

.spec.mysql.pxc.pxc.sidecarResources

Type

object

Property	Type	Description
limits	object	

Property	Type	Description
requests	object	

`.spec.mysql.pxc.pxc.sidecarResources.limits`

Type

object

Property	Type	Description
cpu	string	
ephemeral-storage	string	
memory	string	

`.spec.mysql.pxc.pxc.sidecarResources.requests`

Type

object

Property	Type	Description
cpu	string	
ephemeral-storage	string	
memory	string	

`.spec.mysql.pxc.pxc.startupProbe`

Type

object

Property	Type	Description
<code>failureThreshold</code>	<code>integer</code>	
<code>initialDelaySeconds</code>	<code>integer</code>	
<code>periodSeconds</code>	<code>integer</code>	

`.spec.mysql.pxc.pxc.tolerations`

Type

`array`

`.spec.mysql.pxc.pxc.tolerations[]`

Description

The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.

Type

`object`

Property	Type	Description
<code>effect</code>	<code>string</code>	Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.
<code>key</code>	<code>string</code>	Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is empty, operator must be Exists; this combination means to match all values and all keys.

Property	Type	Description
<code>operator</code>	<code>string</code>	Operator represents a key's relationship to the value. Valid operators are Exists and Equal. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category.
<code>tolerationSeconds</code>	<code>integer</code>	TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is not set, which means tolerate the taint forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.
<code>value</code>	<code>string</code>	Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

`.spec.mysql.pxc.pxc.volumeSpec`

Type

`object`

Property	Type	Description
<code>emptyDir</code>	<code>object</code>	EmptyDir to use as data volume for mysql. EmptyDir represents a temporary directory that shares a pod's lifetime.

Property	Type	Description
<code>hostPath</code>	<code>object</code>	HostPath to use as data volume for mysql. HostPath represents a pre-existing file or directory on the host machine that is directly exposed to the container.
<code>persistentVolumeClaim</code>	<code>object</code>	PersistentVolumeClaim to specify PVC spec for the volume for mysql data. It has the highest level of precedence, followed by HostPath and EmptyDir. And represents the PVC specification.

`.spec.mysql.pxc.pxc.volumeSpec.emptyDir`

Description

EmptyDir to use as data volume for mysql. EmptyDir represents a temporary directory that shares a pod's lifetime.

Type

`object`

Property	Type	Description
<code>medium</code>	<code>string</code>	medium represents what type of storage medium should back this directory. The default is "" which means to use the node's default medium. Must be an empty string (default) or Memory. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir

Property	Type	Description
<code>sizeLimit</code>		<code>sizeLimit</code> is the total amount of local storage required for this EmptyDir volume. The size limit is also applicable for memory medium. The maximum usage on memory medium EmptyDir would be the minimum value between the SizeLimit specified here and the sum of memory limits of all containers in a pod. The default is nil which means that the limit is undefined. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir

.spec.mysql.pxc.pxc.volumeSpec.hostPath

Description

HostPath to use as data volume for mysql. HostPath represents a pre-existing file or directory on the host machine that is directly exposed to the container.

Type

`object`

Required

`path`

Property	Type	Description
<code>path</code>	<code>string</code>	path of the directory on the host. If the path is a symlink, it will follow the link to the real path. More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath
<code>type</code>	<code>string</code>	type for HostPath Volume Defaults to "" More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath

.spec.mysql.pxc.pxc.volumeSpec.persistentVolumeClaim

Description

PersistentVolumeClaim to specify PVC spec for the volume for mysql data. It has the highest level of precedence, followed by HostPath and EmptyDir. And represents the PVC specification.

Type

object

Property	Type	Description
accessModes	array	accessModes contains the desired access modes volume should have. More info: https://kubernetes.io/docs/concepts/storage/persistentvolumes#access-modes-1
dataSource	object	dataSource field can be used to specify either: • An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) • An existing PVC (PersistentVolumeClaim) If the provisioner or an external controller can support the specified data source, it will create a new volume based on the contents of the specified data source. When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If the namespace is specified, then dataSourceRef will not be copied to dataSource.

Property	Type	Description
dataSourceRef	object	dataSourceRef specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a PersistentVolumeClaim object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will require the functionality of the dataSource field and as such both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in dataSourceRef, dataSource isn't set to the same value and must be empty. There are three important differences between dataSource and dataSourceRef • While dataSource only allows two specific type objects, dataSourceRef allows any non-core object as well as PersistentVolumeClaim objects. • While dataSource ignores disallowed values (dropping them), dataSourceRef preserves all values, and generates an error if a disallowed value is specified. • While dataSource only allows local objects, dataSourceRef allows objects in any namespace (Beta) Using this field requires the AnyVolumeDataSource feature gate to be enabled (Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

Property	Type	Description
resources	object	resources represents the minimum resources the volume should have. If RecoverVolumeExpansionFailure feature is enable users are allowed to specify resource requirement that are lower than previous value but must still be higher than capacity recorded in the status field of claim. More info: https://kubernetes.io/docs/concepts/storage/persistentvolumes#resources
selector	object	selector is a label query over volumes to consider binding.
storageClassName	string	storageClassName is the name of the StorageClass required by the claim. More info: https://kubernetes.io/docs/concepts/storage/persistentvolumes#class-1
volumeAttributesClassName	string	volumeAttributesClassName may be used to set the VolumeAttributesClass used by this claim. If specified the CSI driver will create or update the volume with attributes defined in the corresponding VolumeAttributesClass. This has a different purpose than storageClassName, it can be changed after the claim is created. An empty string value means that VolumeAttributesClass will be applied to the claim it's not allowed to reset this field to empty string once it is set. If unspecified and the PersistentVolumeClaim is unbound, the default VolumeAttributesClass will be

Property	Type	Description
		by the persistentvolume controller if it exists. If the resource referred to by volumeAttributesClass does not exist, this PersistentVolumeClaim will be set to Pending state, as reflected by the modifyVolumeStatus field, until such as a resource exists. More info: https://kubernetes.io/docs/concepts/storage/volume-attributes-classes/ (Beta) Using this field requires the VolumeAttributesClass feature gate to be enabled (off by default).
volumeMode	string	volumeMode defines what type of volume is required by the claim. Value of Filesystem is implied when not included in claim spec.
volumeName	string	volumeName is the binding reference to the PersistentVolume backing this claim.

.spec.mysql.pxc.pxc.volumeSpec.persistentVolumeClaim.accessModes

Description

accessModes contains the desired access modes the volume should have. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1>

Type

array

.spec.mysql.pxc.pxc.volumeSpec.persistentVolumeClaim.accessModes[]

Type

string

`.spec.mysql.pxc.pxc.volumeSpec.persistentVolumeClaim.dataSource`

Description

dataSource field can be used to specify either: * An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) * An existing PVC (PersistentVolumeClaim) If the provisioner or an external controller can support the specified data source, it will create a new volume based on the contents of the specified data source. When the AnyVolumeDataSource feature gate is enabled, dataSource contents will be copied to dataSourceRef, and dataSourceRef contents will be copied to dataSource when dataSourceRef.namespace is not specified. If the namespace is specified, then dataSourceRef will not be copied to dataSource.

Type

object

Required

kind

name

Property	Type	Description
<code>apiGroup</code>	<code>string</code>	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
<code>kind</code>	<code>string</code>	Kind is the type of resource being referenced
<code>name</code>	<code>string</code>	Name is the name of resource being referenced

.spec.mysql.pxc.pxc.volumeSpec.persistentVolumeClaim.dataSourceRef

Description

dataSourceRef specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a PersistentVolumeClaim object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replace the functionality of the dataSource field and as such if both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in dataSourceRef, dataSource isn't set to the same value and must be empty. There are three important differences between dataSource and dataSourceRef: * While dataSource only allows two specific types of objects, dataSourceRef allows any non-core object, as well as PersistentVolumeClaim objects. * While dataSource ignores disallowed values (dropping them), dataSourceRef preserves all values, and generates an error if a disallowed value is specified. * While dataSource only allows local objects, dataSourceRef allows objects in any namespaces. (Beta) Using this field requires the AnyVolumeDataSource feature gate to be enabled. (Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

Type

object

Required

kind

name

Property	Type	Description
apiGroup	string	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.

Property	Type	Description
<code>kind</code>	<code>string</code>	Kind is the type of resource being referenced
<code>name</code>	<code>string</code>	Name is the name of resource being referenced
<code>namespace</code>	<code>string</code>	Namespace is the namespace of resource being referenced Note that when a namespace is specified, a gateway.networking.k8s.io/ReferenceGrant object is required in the referent namespace to allow that namespace's owner to accept the reference. See the ReferenceGrant documentation for details. (Alpha) This field requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

`.spec.mysql.pxc.pxc.volumeSpec.persistentVolumeClaim.resources`

Description

`resources` represents the minimum resources the volume should have. If `RecoverVolumeExpansionFailure` feature is enabled users are allowed to specify resource requirements that are lower than previous value but must still be higher than capacity recorded in the status field of the claim. More info: <https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources>

Type

`object`

Property	Type	Description
<code>limits</code>	<code>object</code>	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗
<code>requests</code>	<code>object</code>	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

`.spec.mysql.pxc.pxc.volumeSpec.persistentVolumeClaim.resources.limits`

Description

Limits describes the maximum amount of compute resources allowed. More info:
<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

`object`

`.spec.mysql.pxc.pxc.volumeSpec.persistentVolumeClaim.resources.requests`

Description

Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an

implementation-defined value. Requests cannot exceed Limits. More info:
<https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>

Type

object

.spec.mysql.pxc.pxc.volumeSpec.persistentVolumeClaim.selector

Description

selector is a label query over volumes to consider for binding.

Type

object

Property	Type	Description
<code>matchExpressions</code>	array	matchExpressions is a list of label selector requirements. The requirements are ANDed.
<code>matchLabels</code>	object	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.mysql.pxc.pxc.volumeSpec.persistentVolumeClaim.selector.matchExpressions

Description

matchExpressions is a list of label selector requirements. The requirements are ANDed.

Type

array

`.spec.mysql.pxc.pxc.volumeSpec.persistentVolumeClaim.selector.matchExpressions[]`

Description

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.
operator	string	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
values	array	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

`.spec.mysql.pxc.pxc.volumeSpec.persistentVolumeClaim.selector.matchExpressions[].values`

Description

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

Type

array

.spec.mysql.pxc.pxc.volumeSpec.persistentVolumeClaim.selector.matchExpressions[].values[]

Type

string

.spec.mysql.pxc.pxc.volumeSpec.persistentVolumeClaim.selector.matchLabels

Description

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

Type

object

.spec.mysql.pxc.serviceMonitor

Type

object

Property	Type	Description
customMetricRelabelings	boolean	
enabled	boolean	

Property	Type	Description
<code>interval</code>	<code>string</code>	
<code>labels</code>	<code>object</code>	
<code>metricRelabelings</code>	<code>array</code>	
<code>scrapeTimeout</code>	<code>string</code>	

`.spec.mysql.pxc.serviceMonitor.labels`

Type

`object`

`.spec.mysql.pxc.serviceMonitor.metricRelabelings`

Type

`array`

`.spec.mysql.pxc.serviceMonitor.metricRelabelings[]`

Description

RelabelConfig allows dynamic rewriting of the label set, being applied to samples before ingestion. It defines `<metric_relabel_configs>`-section of Prometheus configuration. More info: https://prometheus.io/docs/prometheus/latest/configuration/configuration/#metric_relabel_config

Type

`object`

Property	Type	Description
<code>action</code>	<code>string</code>	Action to perform based on regex matching. Default is 'replace'

Property	Type	Description
<code>modulus</code>	<code>integer</code>	Modulus to take of the hash of the source label values.
<code>regex</code>	<code>string</code>	Regular expression against which the extracted value is matched. Default is '(.*)'
<code>replacement</code>	<code>string</code>	Replacement value against which a regex replace is performed if the regular expression matches. Regex capture groups are available. Default is '\$1'
<code>separator</code>	<code>string</code>	Separator placed between concatenated source label values. default is ';'.
<code>sourceLabels</code>	<code>array</code>	The source labels select values from existing labels. Their content is concatenated using the configured separator and matched against the configured regular expression for the replace, keep, and drop actions.
<code>targetLabel</code>	<code>string</code>	Label to which the resulting value is written in a replace action. It is mandatory for replace actions. Regex capture groups are available.

`.spec.mysql.pxc.serviceMonitor.metricRelabelings[].sourceLabels`

Description

The source labels select values from existing labels. Their content is concatenated using the configured separator and matched against the configured regular expression for the replace, keep, and drop actions.

Type

array

`.spec.mysql.pxc.serviceMonitor.metricRelabelings[].sourceLabels[]`

Type

string

`.spec.mysql.pxc.slowsql`

Type

object

Property	Type	Description
<code>CKTarget</code>	<code>string</code>	
<code>containerSecurityContext</code>	<code>object</code>	SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.
<code>enabled</code>	<code>boolean</code>	
<code>image</code>	<code>string</code>	

Property	Type	Description
<code>imagePullPolicy</code>	<code>string</code>	PullPolicy describes a policy for if/when to pull a container image
<code>logLevel</code>	<code>string</code>	
<code>pmmServer</code>	<code>object</code>	
<code>resources</code>	<code>object</code>	

.spec.mysql.pxc.slowsql.containerSecurityContext

Description

SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.

Type

`object`

Property	Type	Description
<code>allowPrivilegeEscalation</code>	<code>boolean</code>	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the <code>no_new_privs</code> flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: - run as Privileged - has <code>CAP_SYS_ADMIN</code> Note that this field cannot be set when <code>spec.os.name</code> is windows.

Property	Type	Description
<code>appArmorProfile</code>	<code>object</code>	<code>appArmorProfile</code> is the AppArmor options to use by this container. If set, this profile overrides the pod's <code>appArmorProfile</code> . Note that this field cannot be set when <code>spec.os.name</code> is <code>windows</code> .
<code>capabilities</code>	<code>object</code>	The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when <code>spec.os.name</code> is <code>windows</code> .
<code>privileged</code>	<code>boolean</code>	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to <code>false</code> . Note that this field cannot be set when <code>spec.os.name</code> is <code>windows</code> .
<code>procMount</code>	<code>string</code>	<code>procMount</code> denotes the type of proc mount to use for the containers. The default value is <code>Default</code> which uses the container runtime defaults for readonly paths and masked paths. This requires the <code>ProcMountType</code> feature flag to be enabled. Note that this field cannot be set when <code>spec.os.name</code> is <code>windows</code> .
<code>readOnlyRootFilesystem</code>	<code>boolean</code>	Whether this container has a read-only root filesystem. Default is <code>false</code> . Note that this

Property	Type	Description
		field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

Property	Type	Description
<code>runAsUser</code>	<code>integer</code>	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
<code>seLinuxOptions</code>	<code>object</code>	The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
<code>seccompProfile</code>	<code>object</code>	The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.
<code>windowsOptions</code>	<code>object</code>	The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and

Property	Type	Description
		PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

.spec.mysql.pxc.slowsql.containerSecurityContext.appArmorProfile

Description

appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when spec.os.name is windows.

Type

object

Required

type

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
type	string	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

`.spec.mysql.pxc.slowsql.containerSecurityContext.capabilities`

Description

The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when `spec.os.name` is windows.

Type

object

Property	Type	Description
<code>add</code>	<code>array</code>	Added capabilities
<code>drop</code>	<code>array</code>	Removed capabilities

`.spec.mysql.pxc.slowsql.containerSecurityContext.capabilities.add`

Description

Added capabilities

Type

array

`.spec.mysql.pxc.slowsql.containerSecurityContext.capabilities.add[]`

Description

Capability represent POSIX capabilities type

Type

string

.spec.mysql.pxc.slowsql.containerSecurityContext.capabilities.drop**Description**

Removed capabilities

Type

array

.spec.mysql.pxc.slowsql.containerSecurityContext.capabilities.drop[]**Description**

Capability represent POSIX capabilities type

Type

string

.spec.mysql.pxc.slowsql.containerSecurityContext.seLinuxOptions**Description**

The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Type

object

Property	Type	Description
<code>level</code>	<code>string</code>	Level is SELinux level label that applies to the container.
<code>role</code>	<code>string</code>	Role is a SELinux role label that applies to the container.
<code>type</code>	<code>string</code>	Type is a SELinux type label that applies to the container.
<code>user</code>	<code>string</code>	User is a SELinux user label that applies to the container.

`.spec.mysql.pxc.slowsql.containerSecurityContext.seccompProfile`

Description

The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when `spec.os.name` is windows.

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	localhostProfile indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured

Property	Type	Description
		seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.
		type indicates which kind of seccomp profile will be applied. Valid options are:
type	string	Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied.

.spec.mysql.pxc.slowsql.containerSecurityContext.windowOptions

Description

The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

Type

object

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

.spec.mysql.pxc.slowsql.pmmServer

Type

object

Property	Type	Description
name	string	
namespace	string	

.spec.mysql.pxc.slowsql.resources

Type

object

Property	Type	Description
limits	object	
requests	object	

.spec.mysql.pxc.slowsql.resources.limits

Type

object

Property	Type	Description
cpu	string	
ephemeral-storage	string	
memory	string	

.spec.mysql.pxc.slowsql.resources.requests

Type

object

Property	Type	Description
cpu	string	
ephemeral-storage	string	
memory	string	

.spec.mysql.pxc.tls

Type

object

Property	Type	Description
SANs	array	
issuerConf	object	ObjectReference is a reference to an object with a given name, kind and group.

.spec.mysql.pxc.tls.SANs

Type

array

.spec.mysql.pxc.tls.SANs[]

Type

`string`

`.spec.mysql.pxc.tls.issuerConf`

Description

ObjectReference is a reference to an object with a given name, kind and group.

Type

`object`

Required

`name`

Property	Type	Description
<code>group</code>	<code>string</code>	Group of the resource being referred to.
<code>kind</code>	<code>string</code>	Kind of the resource being referred to.
<code>name</code>	<code>string</code>	Name of the resource being referred to.

`.spec.mysql.pxc.upgradeOptions`

Type

`object`

Property	Type	Description
<code>apply</code>	<code>string</code>	
<code>schedule</code>	<code>string</code>	

Property	Type	Description
<code>versionServiceEndpoint</code>	<code>string</code>	

`.spec.mysql.s3Config`

Description

S3Config is the configuration of S3 for PXC backup

Type

`object`

`.spec.mysql.upgradeOption`

Description

UpgradeOption is the option to upgrade the MySQL cluster

Type

`object`

Property	Type	Description
<code>autoUpgrade</code>	<code>boolean</code>	AutoUpgrade is the flag to auto upgrade the MySQL
<code>crVersion</code>	<code>string</code>	CRVersion is the version of the CR

`.status`

Description

MySQLRestoreStatus defines the observed state of MySQLRestore

Type

`object`

Property	Type	Description
mgr	object	MySQLRestoreStatus defines the observed state of MySQLRestore
state	string	

.status.mgr

Description

MySQLRestoreStatus defines the observed state of MySQLRestore

Type

object

Property	Type	Description
allocated	string	Allocated is the name of pod allocate to.
message	string	Message is the message of run result of restore
progress	integer	Progress is the current restore progress in running.
restoreSchedule	array	RestoreSchedule store this restore schedule to a list.
state	string	State is the overall state for restore

Property	Type	Description
<code>total</code>	<code>integer</code>	Total is the total restore schedule.

`.status.mgr.restoreSchedule`

Description

RestoreSchedule store this restore schedule to a list.

Type

`array`

`.status.mgr.restoreSchedule[]`

Type

`object`

Property	Type	Description
<code>backupInfo</code>	<code>object</code>	BackupInfo is the backup information for restore
<code>message</code>	<code>string</code>	Message is the message of run result of restore
<code>state</code>	<code>string</code>	State is the state for restore

`.status.mgr.restoreSchedule[].backupInfo`

Description

BackupInfo is the backup information for restore

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name is the name of Backup CR.
<code>spec</code>	<code>object</code>	Spec is the spec of Backup CR.
<code>status</code>	<code>object</code>	Status is the status of backup CR.
<code>version</code>	<code>integer</code>	Version is the mark of continues backup.

`.status.mgr.restoreSchedule[].backupInfo.spec`

Description

Spec is the spec of Backup CR.

Type

`object`

Property	Type	Description
<code>cluster</code>	<code>object</code>	Cluster is the name of source cluster.
<code>fullExecutor</code>	<code>string</code>	Executor is the executor of full backup.

Property	Type	Description
<code>mysqlShellDump</code>	<code>object</code>	DumpOption is the dump option for backup.
<code>storage</code>	<code>object</code>	Storage is the storage information of backup for.
<code>storeMeta</code>	<code>boolean</code>	StoreMeta is the flag of store meta data.
<code>type</code>	<code>string</code>	Type means full or increment backup.

`.status.mgr.restoreSchedule[].backupInfo.spec.cluster`

Description

Cluster is the name of source cluster.

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

`.status.mgr.restoreSchedule[].backupInfo.spec.mysqlShellDump`

Description

DumpOption is the dump option for backup.

Type

object

Property	Type	Description
<code>excludeSchemas</code>	array	ExcludeSchemas specifies the schemas to exclude from the backup.
<code>excludeTables</code>	array	ExcludeTables specifies the tables to exclude from the backup, using the 'schema.table' format.
<code>includeSchemas</code>	array	IncludeSchemas specifies the schemas to include in the backup.
<code>includeTables</code>	array	IncludeTables specifies the tables to include in the backup, using the 'schema.table' format.
<code>maxRate</code>	integer	MaxRate specifies the maximum rate at which data is transferred during the backup operation.
<code>threads</code>	integer	Threads specifies the number of threads to use for the backup operation.

.status.mgr.restoreSchedule[].backupInfo.spec.mysqlShellDump.excludeSchemas

Description

ExcludeSchemas specifies the schemas to exclude from the backup.

Type

array

.status.mgr.restoreSchedule[].backupInfo.spec.mysqlShellIDump.excludeSchemas[]

Type

string

.status.mgr.restoreSchedule[].backupInfo.spec.mysqlShellIDump.excludeTables

Description

ExcludeTables specifies the tables to exclude from the backup, using the 'schema.table' format.

Type

array

.status.mgr.restoreSchedule[].backupInfo.spec.mysqlShellIDump.excludeTables[]

Type

string

.status.mgr.restoreSchedule[].backupInfo.spec.mysqlShellIDump.includeSchemas

Description

IncludeSchemas specifies the schemas to include in the backup.

Type

array

.status.mgr.restoreSchedule[].backupInfo.spec.mysqlShellIDump.includeSchemas[]**Type**

string

.status.mgr.restoreSchedule[].backupInfo.spec.mysqlShellIDump.includeTables**Description**

IncludeTables specifies the tables to include in the backup, using the 'schema.table' format.

Type

array

.status.mgr.restoreSchedule[].backupInfo.spec.mysqlShellIDump.includeTables[]**Type**

string

.status.mgr.restoreSchedule[].backupInfo.spec.storage**Description**

Storage is the storage information of backup for.

Type

object

Property	Type	Description
s3	object	S3 means s3 compatible object storage

.status.mgr.restoreSchedule[].backupInfo.spec.storage.s3

Description

S3 means s3 compatible object storage

Type

object

Property	Type	Description
bucket	string	Bucket in which to store the Backup.
endpoint	string	Endpoint (hostname only or fully qualified URI) of S3 compatible storage service.
region	string	Region in which the S3 compatible bucket is located.
secret	object	Secret is a reference to the Secret containing the credentials authenticating with the S3 compatible storage service.

.status.mgr.restoreSchedule[].backupInfo.spec.storage.s3

.secret

Description

Secret is a reference to the Secret containing the credentials authenticating with the S3 compatible storage service.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

.status.mgr.restoreSchedule[].backupInfo.status

Description

Status is the status of backup CR.

Type

object

Property	Type	Description
allocated	string	Allocated is the name of pod allocate to.
dataEndTime	string	DataEndTime is the stop time of increment backup or full backup data.
dataStartTime	string	DataStartTime is the start time of increment backup data.

Property	Type	Description
<code>executor</code>	<code>string</code>	Executor is the executor of backup.
<code>finishTime</code>	<code>string</code>	FinishTime is the finish time of backup runs.
<code>gtidEnd</code>	<code>string</code>	GtidEnd is the stop gtid of backup data
<code>gtidFullPrevious</code>	<code>string</code>	FullGtid is the gtid of full backup if skip a full backup.
<code>gtidPrevious</code>	<code>string</code>	GtidPrevious is the gtid of previous backup.
<code>gtidStart</code>	<code>string</code>	GtidStart is the start gtid of backup data
<code>mark</code>	<code>string</code>	Mark is the mark of backup
<code>memberSize</code>	<code>integer</code>	MemberSize is the size of cluster member.
<code>message</code>	<code>string</code>	Message is the message of run result of backup
<code>path</code>	<code>string</code>	Path is the path of the file stored in storage.

Property	Type	Description
<code>startTime</code>	<code>string</code>	StartTime is the start time of backup runs.
<code>state</code>	<code>string</code>	State is the backup state and enumeration of success, fail or running.
<code>storeMeta</code>	<code>boolean</code>	StoreMeta is the flag of store meta CR.

API Endpoints

The following API endpoints are available:

- `/apis/middleware.alauda.io/v1/namespaces/{namespace}/mysqlrestores`
 - `DELETE` : delete collection of MySQLRestore
 - `GET` : list objects of kind MySQLRestore
 - `POST` : create a new MySQLRestore
- `/apis/middleware.alauda.io/v1/namespaces/{namespace}/mysqlrestores/{name}`
 - `DELETE` : delete the specified MySQLRestore
 - `GET` : read the specified MySQLRestore
 - `PATCH` : partially update the specified MySQLRestore
 - `PUT` : replace the specified MySQLRestore
- `/apis/middleware.alauda.io/v1/namespaces/{namespace}/mysqlrestores/{name}/status`
 - `GET` : read status of the specified MySQLRestore
 - `PATCH` : partially update status of the specified MySQLRestore

- **PUT** : replace status of the specified MySQLRestore

/apis/middleware.alauda.io/v1/namespaces/{namespace}/mysqlrestores

HTTP method

DELETE

Description

delete collection of MySQLRestore

HTTP responses

HTTP code	Response body
200 - OK	Status ✓ schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind MySQLRestore

HTTP responses

HTTP code	Response body
200 - OK	MySQLRestoreList schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new MySQLRestore

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>MySQLRestore</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>MySQLRestore</code> schema
201 - Created	<code>MySQLRestore</code> schema

HTTP code	Response body
202 - Accepted	<code>MySQLRestore</code> schema
401 - Unauthorized	Empty

/apis/middleware.alauda.io/v1/namespaces/{namespace}/mysqlrestores/{name}

HTTP method

`DELETE`

Description

delete the specified MySQLRestore

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	<code>Status ✓</code> schema
202 - Accepted	<code>Status ✓</code> schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified MySQLRestore

HTTP responses

HTTP code	Response body
200 - OK	<code>MySQLRestore</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified MySQLRestore

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are

Parameter	Type	Description
		present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>MySQLRestore</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified MySQLRestore

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a

Parameter	Type	Description
		BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	MySQLRestore schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	MySQLRestore schema
201 - Created	MySQLRestore schema
401 - Unauthorized	Empty

/apis/middleware.alauda.io/v1/namespaces/{namespace}/mysqlrestores/{name}/status

HTTP method

GET

Description

read status of the specified MySQLRestore

HTTP responses

HTTP code	Response body
200 - OK	MySQLRestore schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified MySQLRestore

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>MySQLRestore</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified MySQLRestore

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>MySQLRestore</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>MySQLRestore</code> schema
201 - Created	<code>MySQLRestore</code> schema
401 - Unauthorized	Empty

MGR Schedule

Description

MySQLSchedule is the Schema for the mysqlschedules API

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗

Property	Type	Description
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	MySQLScheduleSpec defines the desired state of MySQLSchedule
status	object	MySQLScheduleStatus defines the observed state of MySQLSchedule

.spec

Description

MySQLScheduleSpec defines the desired state of MySQLSchedule

Type

object

Property	Type	Description
cluster	object	Cluster is the name of source cluster.

Property	Type	Description
<code>expiryDays</code>	<code>integer</code>	ExpiryDays is the days of the latest backup data hold.
<code>full</code>	<code>object</code>	Full is the full backup schedule.
<code>fullExecutor</code>	<code>string</code>	Executor is the executor of full backup.
<code>incr</code>	<code>object</code>	Incr is the increment backup schedule.
<code>mysqlShellDump</code>	<code>object</code>	DumpOption is the dump option for backup.
<code>storage</code>	<code>object</code>	Storage is the storage information of backup for.
<code>storeMeta</code>	<code>boolean</code>	StoreMeta is the flag of store meta data.

.spec.cluster

Description

Cluster is the name of source cluster.

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

.spec.full

Description

Full is the full backup schedule.

Type

`object`

Property	Type	Description
<code>cron</code>	<code>string</code>	Cron is the scheduled backup time expressed using cron.
<code>enable</code>	<code>boolean</code>	Enable is enable or disable schedule backup.

.spec.incr

Description

Incr is the increment backup schedule.

Type

`object`

Property	Type	Description
<code>cron</code>	<code>string</code>	Cron is the scheduled backup time expressed using cron.
<code>enable</code>	<code>boolean</code>	Enable is enable or disable schedule backup.

.spec.mysqlShellDump

Description

DumpOption is the dump option for backup.

Type

`object`

Property	Type	Description
<code>excludeSchemas</code>	<code>array</code>	ExcludeSchemas specifies the schemas to exclude from the backup.
<code>excludeTables</code>	<code>array</code>	ExcludeTables specifies the tables to exclude from the backup, using the 'schema.table' format.
<code>includeSchemas</code>	<code>array</code>	IncludeSchemas specifies the schemas to include in the backup.
<code>includeTables</code>	<code>array</code>	IncludeTables specifies the tables to include in the backup, using the 'schema.table' format.

Property	Type	Description
<code>maxRate</code>	<code>integer</code>	MaxRate specifies the maximum rate at which data is transferred during the backup operation.
<code>threads</code>	<code>integer</code>	Threads specifies the number of threads to use for the backup operation.

`.spec.mysqlShellDump.excludeSchemas`

Description

ExcludeSchemas specifies the schemas to exclude from the backup.

Type

`array`

`.spec.mysqlShellDump.excludeSchemas[]`

Type

`string`

`.spec.mysqlShellDump.excludeTables`

Description

ExcludeTables specifies the tables to exclude from the backup, using the 'schema.table' format.

Type

`array`

`.spec.mysqlShellDump.excludeTables[]`

Type`string`**.spec.mysqlShellDump.includeSchemas****Description**

IncludeSchemas specifies the schemas to include in the backup.

Type`array`**.spec.mysqlShellDump.includeSchemas[]****Type**`string`**.spec.mysqlShellDump.includeTables****Description**

IncludeTables specifies the tables to include in the backup, using the 'schema.table' format.

Type`array`**.spec.mysqlShellDump.includeTables[]****Type**`string`**.spec.storage****Description**

Storage is the storage information of backup for.

Type

object

Property	Type	Description
s3	object	S3 means s3 compatible object storage

.spec.storage.s3

Description

S3 means s3 compatible object storage

Type

object

Property	Type	Description
bucket	string	Bucket in which to store the Backup.
endpoint	string	Endpoint (hostname only or fully qualified URI) of S3 compatible storage service.
region	string	Region in which the S3 compatible bucket is located.
secret	object	Secret is a reference to the Secret containing the credentials authenticating with the S3 compatible storage service.

.spec.storage.s3.secret

Description

Secret is a reference to the Secret containing the credentials authenticating with the S3 compatible storage service.

Type

object

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

.status

Description

MySQLScheduleStatus defines the observed state of MySQLSchedule

Type

object

Property	Type	Description
<code>fullLatestTime</code>	<code>string</code>	FullLatestTime is the full backup latest run time.
<code>incrLatestTime</code>	<code>string</code>	IncrLatestTime is the increment backup latest run time.

API Endpoints

The following API endpoints are available:

- `/apis/mysql.middleware.alauda.io/v1/namespaces/{namespace}/mysqlschedules`
 - **DELETE** : delete collection of MySQLSchedule
 - **GET** : list objects of kind MySQLSchedule
 - **POST** : create a new MySQLSchedule
- `/apis/mysql.middleware.alauda.io/v1/namespaces/{namespace}/mysqlschedules/{name}`
 - **DELETE** : delete the specified MySQLSchedule
 - **GET** : read the specified MySQLSchedule
 - **PATCH** : partially update the specified MySQLSchedule
 - **PUT** : replace the specified MySQLSchedule
- `/apis/mysql.middleware.alauda.io/v1/namespaces/{namespace}/mysqlschedules/{name}/status`
 - **GET** : read status of the specified MySQLSchedule
 - **PATCH** : partially update status of the specified MySQLSchedule
 - **PUT** : replace status of the specified MySQLSchedule

`/apis/mysql.middleware.alauda.io/v1/namespaces/{namespace}/mysqlschedules`

HTTP method

DELETE

Description

delete collection of MySQLSchedule

HTTP responses

HTTP code	Response body
200 - OK	Status ✓ schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind MySQLSchedule

HTTP responses

HTTP code	Response body
200 - OK	<code>MySQLScheduleList</code> schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new MySQLSchedule

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default

Parameter	Type	Description
		in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	MySQLSchedule schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	MySQLSchedule schema
201 - Created	MySQLSchedule schema
202 - Accepted	MySQLSchedule schema
401 - Unauthorized	Empty

/apis/mysql.middleware.alauda.io/v1/namespaces/{namespace}/mysqlschedules/{name}

HTTP method

DELETE

Description

delete the specified MySQLSchedule

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	<code>Status</code> schema
202 - Accepted	<code>Status</code> schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified MySQLSchedule

HTTP responses

HTTP code	Response body
200 - OK	<code>MySQLSchedule</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified MySQLSchedule

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>MySQLSchedule</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified `MySQLSchedule`

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>MySQLSchedule</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>MySQLSchedule</code> schema
201 - Created	<code>MySQLSchedule</code> schema
401 - Unauthorized	Empty

/apis/mysql.middleware.alauda.io/v1/namespaces/{namespace}/mysqlschedules/{name}/status

HTTP method

GET

Description

read status of the specified MySQLSchedule

HTTP responses

HTTP code	Response body
200 - OK	<code>MySQLSchedule</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified MySQLSchedule

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a

Parameter	Type	Description
		warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>MySQLSchedule</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified MySQLSchedule

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last

Parameter	Type	Description
		duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	MySQLSchedule schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	MySQLSchedule schema
201 - Created	MySQLSchedule schema
401 - Unauthorized	Empty



MGR User

Description

MysqlUser is the Schema for the mysqlusers API

Type

object

Specification

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗

Property	Type	Description
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	MysqlUserSpec defines the desired state of MysqlUser
status	object	MysqlUserStatus defines the observed state of MysqlUser

.spec

Description

MysqlUserSpec defines the desired state of MysqlUser

Type

object

Required

host

mysql

user

Property	Type	Description
<code>host</code>	<code>string</code>	Host is the host of the MySQL User
<code>mysql</code>	<code>string</code>	Mysql is the name of the MySQL Cluster
<code>privileges</code>	<code>array</code>	Privileges is the list of MySQL Privileges
<code>secretName</code>	<code>string</code>	SecretName is the name of the Secret for the MySQL User Password
<code>user</code>	<code>string</code>	INSERT ADDITIONAL SPEC FIELDS - desired state of cluster Important: Run "make" to regenerate code after modifying this file User is the name of the MySQL User

`.spec.privileges`

Description

Privileges is the list of MySQL Privileges

Type

`array`

`.spec.privileges[]`

Description

Privilege defines the desired mysql Privilege

Type

`object`

Property	Type	Description
<code>grants</code>	<code>array</code>	Grants is the list of grants for the user
<code>targets</code>	<code>array</code>	Targets is the list target of MySQL Databases or Tables

`.spec.privileges[].grants`

Description

Grants is the list of grants for the user

Type

`array`

`.spec.privileges[].grants[]`

Type

`string`

`.spec.privileges[].targets`

Description

Targets is the list target of MySQL Databases or Tables

Type

`array`

`.spec.privileges[].targets[]`

Type

`string`

.status

Description

MysqlUserStatus defines the observed state of MysqlUser

Type

`object`

Property	Type	Description
<code>lastSyncTime</code>	<code>string</code>	LastSyncTime is the last time the MySQL User sync
<code>reason</code>	<code>string</code>	Reason is the reason of the MySQL User sync failed
<code>secretRef</code>	<code>object</code>	SecretRef is the reference of the Secret for the MySQL User Password
<code>state</code>	<code>string</code>	State is the state of the MySQL User sync

.status.secretRef

Description

SecretRef is the reference of the Secret for the MySQL User Password

Type

`object`

Property	Type	Description
name	string	
namespace	string	

API Endpoints

The following API endpoints are available:

- `/apis/middleware.alauda.io/v1/namespaces/{namespace}/mysqlusers`
 - `DELETE` : delete collection of MysqlUser
 - `GET` : list objects of kind MysqlUser
 - `POST` : create a new MysqlUser
- `/apis/middleware.alauda.io/v1/namespaces/{namespace}/mysqlusers/{name}`
 - `DELETE` : delete the specified MysqlUser
 - `GET` : read the specified MysqlUser
 - `PATCH` : partially update the specified MysqlUser
 - `PUT` : replace the specified MysqlUser
- `/apis/middleware.alauda.io/v1/namespaces/{namespace}/mysqlusers/{name}/status`
 - `GET` : read status of the specified MysqlUser
 - `PATCH` : partially update status of the specified MysqlUser
 - `PUT` : replace status of the specified MysqlUser

`/apis/middleware.alauda.io/v1/namespaces/{namespace}/mysqlusers`

HTTP method

DELETE

Description

delete collection of MysqlUser

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind MysqlUser

HTTP responses

HTTP code	Response body
200 - OK	MysqlUserList schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new MysqlUser

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing

Parameter	Type	Description
		of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>MysqlUser</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>MysqlUser</code> schema
201 - Created	<code>MysqlUser</code> schema
202 - Accepted	<code>MysqlUser</code> schema
401 - Unauthorized	Empty

/apis/middleware.alauda.io/v1/namespaces/{namespace}/mysqlusers/{name}

HTTP method

DELETE

Description

delete the specified MySQLUser

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	<code>Status</code> ✓ schema
202 - Accepted	<code>Status</code> ✓ schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified MySQLUser

HTTP responses

HTTP code	Response body
200 - OK	<code>MySQLUser</code> schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified MySQLUser

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>MysqlUser</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified MysqlUser

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	MysqlUser schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	MysqlUser schema
201 - Created	MysqlUser schema
401 - Unauthorized	Empty

/apis/middleware.alauda.io/v1/namespaces/{namespace}/mysqlusers/{name}/status

HTTP method

GET

Description

read status of the specified MysqlUser

HTTP responses

HTTP code	Response body
200 - OK	MysqlUser schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified MysqlUser

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>MysqlUser</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified `MysqlUser`

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>MysqlUser</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>MysqlUser</code> schema
201 - Created	<code>MysqlUser</code> schema
401 - Unauthorized	Empty

