

# Redis APIs

[Redis](#)

[Redis 用户](#)

[ActiveRedis](#)

# Redis

## Description

Redis is the Schema for the redis API

## Type

object

## Specification

| Property                | Type                | Description  |
|-------------------------|---------------------|--|
| <code>apiVersion</code> | <code>string</code> | <p>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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a></p> |

| 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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a> |
| metadata | ObjectMeta | ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.   |
| spec     | object     | RedisSpec defines the desired state of a Redis instance.   |
| status   | object     | RedisStatus defines the observed state of a Redis instance.  |

## .spec

### Description

RedisSpec defines the desired state of a Redis instance.

### Type

object

### Required

arch

resources

version

| Property                    | Type                | Description   |
|-----------------------------|---------------------|---|
| <code>activeRedis</code>    | <code>object</code> | <p>ActiveRedis configures Redis in active-active replication mode. Set this field to non nil to enable ActiveRedis.</p> <p>NOTE: active-active implementation is not yet complete; only primary-replica mode is supported.</p>  |
| <code>affinity</code>       | <code>object</code> | <p>Affinity defines pod scheduling constraints to control pod placement. See: <a href="https://kubernetes.io/docs/concepts/scheduling-eviction/assign-pod-node/#affinity-and-anti-affinity">https://kubernetes.io/docs/concepts/scheduling-eviction/assign-pod-node/#affinity-and-anti-affinity</a></p> |
| <code>affinityPolicy</code> | <code>string</code> | <p>AffinityPolicy provides predefined affinity policies for Redis pod placement. Options: SoftAntiAffinity, AntiAffinityInSharding, AntiAffinity Default: SoftAntiAffinity</p>  |
| <code>arch</code>           | <code>string</code> | <p>Arch specifies the Redis deployment architecture. Options: cluster, sentinel, standalone</p>   |
| <code>backup</code>         | <code>object</code> | <p>Backup configures automated backup settings for Redis data. Controls backup schedule, retention policy, and storage location.</p>  |

| Property                       | Type                 | Description  |
|--------------------------------|----------------------|--|
| <code>certificate</code>       | <code>object</code>  | Certificate specifies the TLS certificate for Redis when EnableTLS is true.  |
| <code>customConfig</code>      | <code>object</code>  | CustomConfig specifies custom Redis configuration settings that override defaults. These can be set as key-value pairs corresponding to redis.conf parameters. |
| <code>enableActiveRedis</code> | <code>boolean</code> | EnableActiveRedis activates Redis in active-active replication mode. Required for multi-datacenter or multi-region deployments.                                |
| <code>enableTLS</code>         | <code>boolean</code> | EnableTLS enables secure TLS connections for Redis. When enabled, Redis will communicate using encrypted connections.  |
| <code>exporter</code>          | <code>object</code>  | Exporter configures a Redis metrics exporter for Prometheus integration. When enabled, exports Redis metrics in Prometheus format.                             |
| <code>expose</code>            | <code>object</code>  | Expose configures how Redis is exposed outside the Kubernetes cluster. Controls service types, node ports, and load balancer settings.                         |
| <code>ipFamilyPrefer</code>    | <code>string</code>  | IPFamilyPrefer specifies the preferred IP family for Redis networking. Controls whether IPv4 or  |

| Property                    | Type                | Description  |
|-----------------------------|---------------------|--|
|                             |                     | IPv6 is used for pod and service communication.  |
| <code>modules</code>        | <code>array</code>  | Modules defines a list of Redis modules to be loaded into the Redis instance. Each module is specified by its name and version. Modules are loaded at startup and can extend Redis functionality.  |
| <code>nodeSelector</code>   | <code>object</code> | NodeSelector constrains Redis pods to nodes with matching labels. See: <a href="https://kubernetes.io/docs/concepts/scheduling-eviction/assign-pod-node/#nodeselector">https://kubernetes.io/docs/concepts/scheduling-eviction/assign-pod-node/#nodeselector</a> |
| <code>password</code>       | <code>string</code> | Password is the authentication credential for Redis. Deprecated: Use PasswordSecret instead for improved security.   |
| <code>passwordSecret</code> | <code>string</code> | PasswordSecret references a Kubernetes secret containing the Redis password. The secret must have a 'password' key with a value matching: <code>^[a-zA-Z0-9_!@#\$\$%^&amp;*()-_+=+?]{8,32}\$</code>  |
| <code>patches</code>        | <code>object</code> | Patches allows customization of generated Kubernetes resources. Enables fine-grained control over services and other child resources.  |

| Property                    | Type                 | Description  |
|-----------------------------|----------------------|--|
| <code>pause</code>          | <code>boolean</code> | Pause temporarily stops Redis reconciliation when set to true. Useful for maintenance operations or debugging.   |
| <code>persistent</code>     | <code>object</code>  | Persistent configures persistent storage for Redis data. If specified, a PersistentVolumeClaim will be created for each Redis node.                    |
| <code>persistentSize</code> | <code>object</code>  | PersistentSize specifies the size of the persistent volume for each Redis node. Only applicable when Persistent is configured.                         |
| <code>podAnnotations</code> | <code>object</code>  | PodAnnotations are custom annotations to add to all Redis pods. Useful for integrations with service meshes, monitoring tools, etc.                    |
| <code>redisProxy</code>     | <code>object</code>  | RedisProxy configures an optional proxy layer in front of Redis. Deprecated: Will be removed in 3.20.  |
| <code>replicas</code>       | <code>object</code>  | Replicas defines the number of Redis nodes for different architectures. Configuration varies based on the selected architecture (cluster or sentinel). |

| Property                          | Type                 | Description   |
|-----------------------------------|----------------------|---|
| <code>resources</code>            | <code>object</code>  | Resources defines the compute resource requirements for Redis containers. This includes CPU and memory limits/requests.                               |
| <code>restore</code>              | <code>object</code>  | Restore configures data restoration from a backup. Specifies the backup source and restoration parameters.  |
| <code>securityContext</code>      | <code>object</code>  | SecurityContext defines security settings for Redis pods, including user/group IDs, filesystem permissions, and SELinux context.                      |
| <code>sentinel</code>             | <code>object</code>  | Sentinel configures Redis Sentinel for high availability. Only applicable when arch is set to "sentinel".   |
| <code>sentinelCustomConfig</code> | <code>object</code>  | SentinelCustomConfig defines custom Redis Sentinel configuration settings. Deprecated: Use sentinel.customConfig instead for improved organization.   |
| <code>serviceID</code>            | <code>integer</code> | ServiceID specifies the unique identifier for this Redis instance in an active-active setup. Must be unique across all Active-Active Redis instances. |

| Property                   | Type                | Description  |
|----------------------------|---------------------|--|
| <code>tolerations</code>   | <code>array</code>  | Tolerations allow Redis pods to be scheduled on nodes with matching taints. See: <a href="https://kubernetes.io/docs/concepts/scheduling-eviction/taint-and-toleration/">https://kubernetes.io/docs/concepts/scheduling-eviction/taint-and-toleration/</a> |
| <code>upgradeOption</code> | <code>object</code> | UpgradeOption defines the upgrade strategy and automation settings. Controls how and when Redis instances are upgraded.  |
| <code>version</code>       | <code>string</code> | Version specifies the Redis version to deploy. Currently supports: 5.0, 6.0, 6.2, 7.0, 7.2, 7.4  |

## .spec.activeRedis

### Description

ActiveRedis configures Redis in active-active replication mode. Set this field to non nil to enable ActiveRedis. NOTE: active-active implementation is not yet complete; only primary-replica mode is supported.

### Type

`object`

| Property           | Type                | Description  |
|--------------------|---------------------|--|
| <code>proxy</code> | <code>object</code> | Proxy configures the Redis Proxy settings for active-active replication. This includes proxy image, replicas, and resource requirements. |

| Property               | Type                 | Description   |
|------------------------|----------------------|---|
| <code>serviceID</code> | <code>integer</code> | ServiceID specifies the unique identifier for this Redis instance in an active-active setup. Must be unique across all Active-Active Redis instances. |

## .spec.activeRedis.proxy

### Description

Proxy configures the Redis Proxy settings for active-active replication. This includes proxy image, replicas, and resource requirements.

### Type

`object`

| Property                              | Type                | Description  |
|---------------------------------------|---------------------|--|
| <code>affinity</code>                 | <code>object</code> | Affinity is the affinity for activeredis proxy                   |
| <code>containerSecurityContext</code> | <code>object</code> | ContainerSecurityContext   |
| <code>image</code>                    | <code>string</code> | Image is the proxy image   |
| <code>imagePullPolicy</code>          | <code>string</code> | ImagePullPolicy is the image pull policy for activeredis proxy   |
| <code>imagePullSecrets</code>         | <code>array</code>  | ImagePullSecrets is the image pull secrets for activeredis proxy |

| Property                     | Type                 | Description   |
|------------------------------|----------------------|---|
| <code>nodeSelector</code>    | <code>object</code>  | NodeSelector is the node selector for activeredis proxy |
| <code>replicas</code>        | <code>integer</code> | ServiceID the service id for activeredis                |
| <code>resources</code>       | <code>object</code>  | Resources is the resource for activeredis proxy         |
| <code>securityContext</code> | <code>object</code>  | SecurityContext   |
| <code>service</code>         | <code>object</code>  | Service is the service for activeredis proxy            |
| <code>tolerations</code>     | <code>array</code>   | Tolerations is the tolerations for activeredis proxy    |

## `.spec.activeRedis.proxy.affinity`

### Description

Affinity is the affinity for activeredis proxy

### 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.activeRedis.proxy.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 |

| Property   | Type                       | Description  |
|--|----------------------------|--|
|  |                            | <p>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 adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest sum are the most preferred.</p> |
| <p><code>requiredDuringSchedulingIgnoredDuringExecution</code></p> | <p><code>object</code></p> | <p>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.</p>   |

## `.spec.activeRedis.proxy.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.activeRedis.proxy.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 <code>nodeSelectorTerm</code> , in the range 1-100. |

## `.spec.activeRedis.proxy.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.activeRedis.proxy.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions`

### Description

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

### Type

`array`

## `.spec.activeRedis.proxy.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.activeRedis.proxy.affinity.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.activeRedis.proxy.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values[]**

### Type

string

## **.spec.activeRedis.proxy.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields**

### Description

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

### Type

array

## **.spec.activeRedis.proxy.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.activeRedis.proxy.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.activeRedis.proxy.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values[]**

**Type**

string

**.spec.activeRedis.proxy.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.activeRedis.proxy.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms**

## Description

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

## Type

array

# **.spec.activeRedis.proxy.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  |
|------------------|-------|--|
| 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.activeRedis.proxy.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[ ].matchExpressions**

## Description

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

## Type

array

# `.spec.activeRedis.proxy.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.  |
| 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.activeRedis.proxy.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.activeRedis.proxy.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values[]`

### Type

string

## `.spec.activeRedis.proxy.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields`

### Description

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

### Type

array

## `.spec.activeRedis.proxy.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.activeRedis.proxy.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.activeRedis.proxy.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values[]**

## Type

string

**.spec.activeRedis.proxy.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, it may choose a node that violates one or more 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 o |

| Property   | Type                      | Description  |
|--|---------------------------|--|
|  |                           | <p>scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution, affinity expressions, etc.) to compute a sum by iterating through the elements of this field adding "weight" to the sum if the node has a label which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.</p>  |
| <p><code>requiredDuringSchedulingIgnoredDuringExecution</code></p> | <p><code>array</code></p> | <p>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</p> |

| Property | Type | Description                                    |
|----------|------|--|
|          |      | intersected, i.e. all terms must be satisfied. |

## `.spec.activeRedis.proxy.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.activeRedis.proxy.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.activeRedis.proxy.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 <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 <code>labelSelector</code> as <code>key in (value)</code> to select |

| Property                       | Type                | Description   |
|--------------------------------|---------------------|---|
|                                |                     | <p>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.</p>   |
| <code>mismatchLabelKeys</code> | <code>array</code>  | <p><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.</p> |
| <code>namespaceSelector</code> | <code>object</code> | <p>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.</p>  |
|                                |                     |   |

| 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.activeRedis.proxy.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>  | <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.activeRedis.proxy.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions`

### Description

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

### Type

`array`

## `.spec.activeRedis.proxy.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.activeRedis.proxy.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.activeRedis.proxy.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values**

## labelSelector.matchExpressions[].values[]

### Type

string

## .spec.activeRedis.proxy.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.activeRedis.proxy.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.

### Type

array

**.spec.activeRedis.proxy.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]**

Type

string

**.spec.activeRedis.proxy.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.

Type

array

**.spec.activeRedis.proxy.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]**

Type

string

**.spec.activeRedis.proxy.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.name**

# 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.activeRedis.proxy.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces.namespaceSelector.matchExpressions

## Description

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

## Type

array

## `.spec.activeRedis.proxy.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.   |
| 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.activeRedis.proxy.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.activeRedis.proxy.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]**

#### Type

string

**.spec.activeRedis.proxy.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.activeRedis.proxy.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.activeRedis.proxy.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]**

### Type

string

## **.spec.activeRedis.proxy.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.activeRedis.proxy.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>     | 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. |
| <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 not in</code>  |

| Property                       | Type                | Description  |
|--------------------------------|---------------------|--|
|                                |                     | <p><code>(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.</p> |
| <code>namespaceSelector</code> | <code>object</code> | <p>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.</p>  |
| <code>namespaces</code>        | <code>array</code>  | <p><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".</p>   |
|                                |                     |  |

| 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.activeRedis.proxy.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>  | <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.activeRedis.proxy.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

### Description

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

### Type

array

## `.spec.activeRedis.proxy.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. |
|          |        |  |

| 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.activeRedis.proxy.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.activeRedis.proxy.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]**

### Type

string

## **.spec.activeRedis.proxy.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.activeRedis.proxy.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.

### Type

array

## **.spec.activeRedis.proxy.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]**

### Type

string

## **.spec.activeRedis.proxy.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.

### Type

array

## `.spec.activeRedis.proxy.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys`

### Type

string

## `.spec.activeRedis.proxy.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 | <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.activeRedis.proxy.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions`

### Description

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

### Type

`array`

## `.spec.activeRedis.proxy.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.activeRedis.proxy.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.activeRedis.proxy.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]**

### **Type**

`string`

## **.spec.activeRedis.proxy.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.activeRedis.proxy.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.activeRedis.proxy.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]**

### **Type**

`string`

## **.spec.activeRedis.proxy.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, 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 a podAffinityTerm which matches the corresponding podAffinityTerm; the node(s) with the highest sum is the most preferred. |

| Property  | Type  | Description   |
|---|-------|---|
|   |       | sum are the most preferred.   |
| <code>requiredDuringSchedulingIgnoredDuringExecution</code> | array | If the anti-affinity requirements specified by this field are not met at the time of scheduling, the pod will not be scheduled on 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 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 intersected, i.e. all terms must be satisfied. |

## `.spec.activeRedis.proxy.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.activeRedis.proxy.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  |
|-----------------|---------|--|
| 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.activeRedis.proxy.affinity.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 <code>matchLabelKeys</code> and <code>labelSelector</code> . Also, <code>matchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. |
| <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  |

| Property                       | Type                | Description  |
|--------------------------------|---------------------|--|
|                                |                     | 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. |
| <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  |

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

## `.spec.activeRedis.proxy.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.activeRedis.proxy.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions`

## Description

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

## Type

array

# `.spec.activeRedis.proxy.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.activeRedis.proxy.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.activeRedis.proxy.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]`

### Type

string

## `.spec.activeRedis.proxy.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.activeRedis.proxy.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.

### Type

array

## `.spec.activeRedis.proxy.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]`

### Type

string

## `.spec.activeRedis.proxy.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.

### Type

array

## `.spec.activeRedis.proxy.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]`

### Type

string

## `.spec.activeRedis.proxy.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  | <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 |

| Property | Type | Description   |
|----------|------|---|
|          |      | 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.activeRedis.proxy.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions`

### Description

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

### Type

array

## `.spec.activeRedis.proxy.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.activeRedis.proxy.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.activeRedis.proxy.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]**

### Type

`string`

## `.spec.activeRedis.proxy.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.activeRedis.proxy.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.activeRedis.proxy.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]`

### Type

`string`

## .spec.activeRedis.proxy.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.activeRedis.proxy.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>  | <p>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.</p>              |
| <code>mismatchLabelKeys</code> | <code>array</code>  | <p>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 not 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>mismatchLabelKeys</code> and <code>labelSelector</code>. Also, <code>mismatchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set.</p> |
| <code>namespaceSelector</code> | <code>object</code> | <p>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</p>  |

| Property    | Type   | Description  |
|-------------|--------|--|
|             |        | 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.activeRedis.proxy.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> | <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.activeRedis.proxy.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

### Description

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

### Type

`array`

## `.spec.activeRedis.proxy.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.activeRedis.proxy.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.activeRedis.proxy.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.`

## matchExpressions[].values[]

### Type

string

## .spec.activeRedis.proxy.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.activeRedis.proxy.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.

### Type

array

## `.spec.activeRedis.proxy.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]`

### Type

string

## `.spec.activeRedis.proxy.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.

### Type

array

## `.spec.activeRedis.proxy.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]`

### Type

string

## `.spec.activeRedis.proxy.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSel`

## ector

### 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.activeRedis.proxy.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions

### Description

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

### Type

array

## `.spec.activeRedis.proxy.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.activeRedis.proxy.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.activeRedis.proxy.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]**

#### Type

string

**.spec.activeRedis.proxy.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.activeRedis.proxy.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.activeRedis.proxy.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]****Type**

string

**.spec.activeRedis.proxy.containerSecurityContext****Description**

ContainerSecurityContext

**Type**

object

| Property                 | Type    | Description  |
|--------------------------|---------|--|
| allowPrivilegeEscalation | boolean | <p>AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the no_new_privs flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is:</p> <ol style="list-style-type: none"> <li>run as Privileged</li> <li>has CAP_SYS_ADMIN Note that this field cannot be set when spec.os.name is windows.</li> </ol> |
| appArmorProfile          | object  | appArmorProfile is the AppArmor options to use by this container. If set, this profile   |

| Property                            | Type                 | Description  |
|-------------------------------------|----------------------|--|
|                                     |                      | overrides the pod's appArmorProfile. Note that this field cannot be set when spec.os.name 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 spec.os.name 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 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.   |

| Property                  | Type                 | Description   |
|---------------------------|----------------------|---|
| <code>runAsGroup</code>   | <code>integer</code> | <p>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.</p>   |
| <code>runAsNonRoot</code> | <code>boolean</code> | <p>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.</p> |
| <code>runAsUser</code>    | <code>integer</code> | <p>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.</p>   |

| Property                    | Type                | Description  |
|-----------------------------|---------------------|--|
| <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 PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.  |

**.spec.activeRedis.proxy.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.activeRedis.proxy.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.activeRedis.proxy.containerSecurityContext.capabilities.add**

### **Description**

Added capabilities

### **Type**

array

## **.spec.activeRedis.proxy.containerSecurityContext.capabilities.add[]**

### **Description**

Capability represent POSIX capabilities type

### **Type**

string

## **.spec.activeRedis.proxy.containerSecurityContext.capabilities.drop**

### **Description**

Removed capabilities

## Type

array

# .spec.activeRedis.proxy.containerSecurityContext.capabilities.drop[]

## Description

Capability represent POSIX capabilities type

## Type

string

# .spec.activeRedis.proxy.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. |
|          |        |   |

| 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.activeRedis.proxy.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> | <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 <code>type</code> 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:   |

| Property | Type | Description   |
|----------|------|---|
|          |      | 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.activeRedis.proxy.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 ( <a href="https://github.com/kubernetes-sigs/windows-gmsa">https://github.com/kubernetes-sigs/windows-gmsa</a> ↗) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field. |
| <code>gmsaCredentialSpecName</code> | <code>string</code> | <code>GMSACredentialSpecName</code> is the name of the GMSA credential spec to use.  |

| Property      | Type    | Description  |
|---------------|---------|--|
| 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. |
| runAsUserName | string  | 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.activeRedis.proxy.imagePullSecrets`

### Description

ImagePullSecrets is the image pull secrets for activeredis proxy

### Type

array

## `.spec.activeRedis.proxy.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: <a href="https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names">https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names</a> |

## .spec.activeRedis.proxy.nodeSelector

### Description

NodeSelector is the node selector for activeredis proxy

### Type

object

## .spec.activeRedis.proxy.resources

### Description

Resources is the resource for activeredis proxy

### Type

object

| Property | Type  | Description   |
|----------|-------|---|
| claims   | array | Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. |

| Property | Type   | Description  |
|----------|--------|--|
|          |        | <p>This is an alpha field and requires enabling the DynamicResourceAllocation feature gate.</p> <p>This field is immutable. It can only be set for containers.</p>   |
| limits   | object | <p>Limits describes the maximum amount of compute resources allowed. More info: <a href="https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/">https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/</a> ↗</p>  |
| requests | object | <p>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: <a href="https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/">https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/</a> ↗</p> |

## `.spec.activeRedis.proxy.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.activeRedis.proxy.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.activeRedis.proxy.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.activeRedis.proxy.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.activeRedis.proxy.securityContext

## Description

SecurityContext

## Type

object

| Property                     | Type   | Description  |
|------------------------------|--------|--|
| <code>appArmorProfile</code> | 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. |

| Property                         | Type                 | Description  |
|----------------------------------|----------------------|--|
| <code>fsGroup</code>             | <code>integer</code> | <p>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:</p> <ol style="list-style-type: none"> <li>1. The owning GID will be the FSGroup</li> <li>2. The setgid bit is set (new files created in the volume will be owned by FSGroup)</li> <li>3. The permission bits are OR'd with rw-rw----</li> </ol> <p>If unset, the Kubelet will not modify the ownership and permissions of any volume. Note that this field cannot be set when <code>spec.os.name</code> is windows.</p> |
| <code>fsGroupChangePolicy</code> | <code>string</code>  | <p><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 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 <code>spec.os.name</code> is windows.</p>   |

| Property                  | Type                 | Description  |
|---------------------------|----------------------|--|
| <code>runAsGroup</code>   | <code>integer</code> | <p>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.</p>  |
| <code>runAsNonRoot</code> | <code>boolean</code> | <p>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.</p> |
| <code>runAsUser</code>    | <code>integer</code> | <p>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.</p>  |

| Property                         | Type                | Description  |
|----------------------------------|---------------------|--|
| <code>seLinuxChangePolicy</code> | <code>string</code> | <p><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".</p> <p>"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.</p> <p>"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.</p> <p>"MountOption" value is allowed only when SELinuxMount feature gate is enabled.</p> <p>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.</p> |

| Property           | Type   | Description  |
|--------------------|--------|--|
|                    |        | <p>This field affects only Pods that have SELinux label set, either in PodSecurityContext or in SecurityContext of all containers.</p> <p>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.</p>  |
| seLinuxOptions     | object | <p>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.</p> |
| seccompProfile     | object | <p>The seccomp options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.</p>  |
| supplementalGroups | array  | <p>A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if</p>   |

| Property                              | Type                | Description   |
|---------------------------------------|---------------------|---|
|                                       |                     | 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.   |
| <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.   |
|                                       |                     |   |

| Property                    | Type                | Description   |
|-----------------------------|---------------------|---|
| <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 <code>spec.os.name</code> is linux. |

## `.spec.activeRedis.proxy.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> | <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". |
| <code>type</code>             | <code>string</code> | <code>type</code> 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.activeRedis.proxy.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   |
|----------|--------|---|
| 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.activeRedis.proxy.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> | string | <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>             | string | <code>type</code> indicates which kind of seccomp profile will be applied. Valid options are:<br><br>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.activeRedis.proxy.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.activeRedis.proxy.securityContext.supplementalGroups[]****Type**

integer

**.spec.activeRedis.proxy.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.activeRedis.proxy.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.activeRedis.proxy.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 ( <a href="https://github.com/kubernetes-sigs/windows-gmsa">https://github.com/kubernetes-sigs/windows-gmsa</a> ^) inlines the contents of the GMSA credential spec named by the GMSACredentialSpecName field. |
| gmsaCredentialSpecName | string | GMSACredentialSpecName is the name of the GMSA credential spec to use.  |

| Property                   | Type                 | Description  |
|----------------------------|----------------------|--|
| <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.activeRedis.proxy.service`

### Description

Service is the service for activeredis proxy

### Type

`object`

| Property                 | Type                | Description                                |
|--------------------------|---------------------|--|
| <code>annotations</code> | <code>object</code> | Annotations is the annotations for service |

| Property               | Type                 | Description  |
|------------------------|----------------------|--|
| <code>enableTLS</code> | <code>boolean</code> | EnableTLS is the enable tls for activeredis proxy  |
| <code>port</code>      | <code>integer</code> | Port custom nodeport for activeredis proxy service |
| <code>type</code>      | <code>string</code>  | Type is the type of service                        |

## `.spec.activeRedis.proxy.service.annotations`

### Description

Annotations is the annotations for service

### Type

`object`

## `.spec.activeRedis.proxy.tolerations`

### Description

Tolerations is the tolerations for activeredis proxy

### Type

`array`

## `.spec.activeRedis.proxy.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 forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system. |
| value             | string  | 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.affinity

### Description

Affinity defines pod scheduling constraints to control pod placement. See: <https://kubernetes.io/docs/concepts/scheduling-eviction/assign-pod-node/#affinity-and-anti-affinity>

### 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.affinity.nodeAffinity

### Description

Describes node affinity scheduling rules for the pod.

### Type

object

| Property  | Type          | Description   |
|---|---------------|---|
| <p><code>preferredDuringSchedulingIgnoredDuringExecution</code></p> | <p>array</p>  | <p>The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field. 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.</p> |
| <p><code>requiredDuringSchedulingIgnoredDuringExecution</code></p>  | <p>object</p> | <p>If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node.</p>  |

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

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

### Description

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

### Type

array

## `.spec.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 |

| Property | Type | Description  |
|----------|------|--|
|          |      | element, which will be interpreted as an integer. This array is replaced during a strategic merge patch. |

## **.spec.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.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values[]**

### Type

string

## **.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields**

### Description

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

### Type

array

# `.spec.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.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.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values[]**

#### Type

string

## **.spec.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.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms`

### Description

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

### Type

array

## `.spec.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.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions`

### Description

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

## Type

array

# .spec.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.  |
| 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.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.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values[]`

### Type

string

## `.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields`

### Description

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

### Type

array

## `.spec.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.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.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values[]****Type**

string

**.spec.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  |
|---|-------|--|
| preferredDuringSchedulingIgnoredDuringExecution | array | The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, 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 o |

| Property   | Type                      | Description  |
|--|---------------------------|--|
|  |                           | <p>scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution, affinity expressions, etc.) to compute a sum by iterating through the elements of this field adding "weight" to the sum if the node has a label which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.</p>  |
| <p><code>requiredDuringSchedulingIgnoredDuringExecution</code></p> | <p><code>array</code></p> | <p>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</p> |

| Property | Type | Description                                    |
|----------|------|--|
|          |      | intersected, i.e. all terms must be satisfied. |

## `.spec.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.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.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 <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 <code>labelSelector</code> as <code>key in (value)</code> to select |

| Property                       | Type                | Description   |
|--------------------------------|---------------------|---|
|                                |                     | <p>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.</p>   |
| <code>mismatchLabelKeys</code> | <code>array</code>  | <p><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.</p> |
| <code>namespaceSelector</code> | <code>object</code> | <p>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.</p>  |
|                                |                     |   |

| 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.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. |

| 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.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions`

### Description

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

### Type

`array`

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

### Type

`string`

## `.spec.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.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.

### Type

`array`

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

### Type

`string`

## `.spec.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.

### Type

array

## `.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys` `[]`

### Type

string

## `.spec.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.   |
| <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.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions`

### Description

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

### Type

array

## `.spec.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.   |
| 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.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.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.**

## matchExpressions[].values[]

### Type

string

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

### Type

string

## .spec.affinity.podAffinity.requiredDuringSchedulingIgnore dDuringExecution

### 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.affinity.podAffinity.requiredDuringSchedulingIgnore dDuringExecution[]

### 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>  | <p>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.</p>              |
| <code>mismatchLabelKeys</code> | <code>array</code>  | <p>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 not 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>mismatchLabelKeys</code> and <code>labelSelector</code>. Also, <code>mismatchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set.</p> |
| <code>namespaceSelector</code> | <code>object</code> | <p>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</p>  |

| Property    | Type   | Description  |
|-------------|--------|--|
|             |        | 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.affinity.podAffinity.requiredDuringSchedulingIgnore` `dDuringExecution[].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.affinity.podAffinity.requiredDuringSchedulingIgnoreDuringExecution[].labelSelector.matchExpressions`

### Description

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

### Type

`array`

## `.spec.affinity.podAffinity.requiredDuringSchedulingIgnoreDuringExecution[].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.affinity.podAffinity.requiredDuringSchedulingIgnore dDuringExecution[].labelSelector.matchExpressions[].val ues**

### **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.affinity.podAffinity.requiredDuringSchedulingIgnore dDuringExecution[].labelSelector.matchExpressions[].val ues[]**

### **Type**

`string`

## `.spec.affinity.podAffinity.requiredDuringSchedulingIgnore dDuringExecution[].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.affinity.podAffinity.requiredDuringSchedulingIgnore dDuringExecution[].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.

### Type

`array`

## `.spec.affinity.podAffinity.requiredDuringSchedulingIgnore dDuringExecution[].matchLabelKeys[]`

### Type

`string`

## `.spec.affinity.podAffinity.requiredDuringSchedulingIgnore dDuringExecution[].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.

### Type

array

## `.spec.affinity.podAffinity.requiredDuringSchedulingIgnore dDuringExecution[].mismatchLabelKeys[]`

### Type

string

## `.spec.affinity.podAffinity.requiredDuringSchedulingIgnore dDuringExecution[].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.affinity.podAffinity.requiredDuringSchedulingIgnore dDuringExecution[].namespaceSelector.matchExpression s`

### Description

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

### Type

`array`

## `.spec.affinity.podAffinity.requiredDuringSchedulingIgnore dDuringExecution[].namespaceSelector.matchExpression s[]`

### 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.affinity.podAffinity.requiredDuringSchedulingIgnore dDuringExecution[].namespaceSelector.matchExpression s[].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.affinity.podAffinity.requiredDuringSchedulingIgnore dDuringExecution[].namespaceSelector.matchExpression`

## s[].values[]

### Type

string

## .spec.affinity.podAffinity.requiredDuringSchedulingIgnore dDuringExecution[].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.affinity.podAffinity.requiredDuringSchedulingIgnore dDuringExecution[].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.affinity.podAffinity.requiredDuringSchedulingIgnore dDuringExecution[].namespaces[]

### Type

string

# .spec.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 expression specified by this field. 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 expression etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has a pod which matches the corresponding podAffinityTerm; the |

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

## `.spec.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.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  |
|-----------------|---------|--|
| 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.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 <code>matchLabelKeys</code> and <code>labelSelector</code> . Also, <code>matchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. |
| 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 not in</code>  |

| Property                       | Type                | Description  |
|--------------------------------|---------------------|--|
|                                |                     | <p><code>(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.</p> |
| <code>namespaceSelector</code> | <code>object</code> | <p>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.</p>  |
| <code>namespaces</code>        | <code>array</code>  | <p><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".</p>   |
|                                |                     |  |

| 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.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.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions`

### Description

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

### Type

array

## `.spec.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. |
|          |        |  |

| Property            | Type               | Description  |
|---------------------|--------------------|--|
| <code>values</code> | <code>array</code> | <p><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.</p> |

## `.spec.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.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]`

### Type

`string`

## `.spec.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.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.

### Type

array

## **.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys**

[]

### Type

string

## **.spec.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.

#### Type

array

## `.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]`

#### Type

string

## `.spec.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  | <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.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions`

### Description

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

### Type

array

## `.spec.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.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.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values**

## `tor.matchExpressions[].values[]`

### Type

string

## `.spec.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.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.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]`

### Type

string

## .spec.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.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>  | <p>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.</p>              |
| <code>mismatchLabelKeys</code> | <code>array</code>  | <p>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 not 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>mismatchLabelKeys</code> and <code>labelSelector</code>. Also, <code>mismatchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set.</p> |
| <code>namespaceSelector</code> | <code>object</code> | <p>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</p>  |

| Property    | Type   | Description  |
|-------------|--------|--|
|             |        | 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.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> | <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.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

### Description

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

### Type

`array`

## `.spec.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.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.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]**

#### Type

`string`

## `.spec.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.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.

### Type

`array`

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

### Type

`string`

## `.spec.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.

### Type

array

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

### Type

string

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

### Description

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

### Type

array

## `.spec.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.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.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpres`

## sions[].values[]

### Type

string

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

### Type

string

## .spec.backup

### Description

Backup configures automated backup settings for Redis data. Controls backup schedule, retention policy, and storage location.

### Type

object

| Property | Type   | Description |
|----------|--------|-------------|
| image    | string |             |
| schedule | array  |             |

## .spec.backup.schedule

### Type

array

## .spec.backup.schedule[]

### Type

object

### Required

keep

schedule

storage

| Property          | Type    | Description |
|-------------------|---------|-------------|
| keep              | integer |             |
| keepAfterDeletion | boolean |             |
| name              | string  |             |

| Property | Type   | Description |
|----------|--------|-------------|
| schedule | string |             |
| storage  | object |             |
| target   | object |             |

## .spec.backup.schedule[].storage

### Type

object

| Property         | Type   | Description |
|------------------|--------|-------------|
| size             |        |             |
| storageClassName | string |             |

## .spec.backup.schedule[].target

### Type

object

| Property | Type   | Description |
|----------|--------|-------------|
| s3option | object | S3Option    |

## .spec.backup.schedule[].target.s3Option

### Description

S3Option

### Type

object

| Property | Type   | Description |
|----------|--------|-------------|
| bucket   | string |             |
| dir      | string |             |
| s3Secret | string |             |

## .spec.certificate

### Description

Certificate specifies the TLS certificate for Redis when EnableTLS is true.

### Type

object

| Property | Type   | Description  |
|----------|--------|--|
| dnsNames | array  | DNSNames is the list of DNS names to be included in the certificate  |
| duration | string | Requested 'duration' (i.e. lifetime) of the Certificate. Note that the issuer may choose to ignore the requested duration, just like any other requested attribute.<br><br>If unset, this defaults to 10 years. Minimum accepted duration is 1 hour. Value must be in units accepted by Go time.ParseDuration<br><a href="https://golang.org/pkg/time/#ParseDuration">https://golang.org/pkg/time/#ParseDuration</a> . |
| issuer   | object | Issuer is the issuer of the certificate  |

## `.spec.certificate.dnsNames`

### Description

DNSNames is the list of DNS names to be included in the certificate

### Type

array

## `.spec.certificate.dnsNames[]`

### Type

string

## `.spec.certificate.issuer`

### Description

Issuer is the issuer of the certificate

### Type

object

| Property           | Type                | Description         |
|--------------------|---------------------|---------------------|
| <code>group</code> | <code>string</code> | Group of the Issuer |
| <code>kind</code>  | <code>string</code> | Kind of the Issuer  |
| <code>name</code>  | <code>string</code> | Name of the Issuer  |

## `.spec.customConfig`

## Description

CustomConfig specifies custom Redis configuration settings that override defaults. These can be set as key-value pairs corresponding to redis.conf parameters.

## Type

object

## .spec.exporter

### Description

Exporter configures a Redis metrics exporter for Prometheus integration. When enabled, exports Redis metrics in Prometheus format.

### Type

object

| Property                     | Type                 | Description   |
|------------------------------|----------------------|---|
| <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>  | ResourceRequirements describes the compute resource requirements.   |

## .spec.exporter.resources

### Description

ResourceRequirements describes the compute resource requirements.

### Type

object

| Property | Type   | Description  |
|----------|--------|--|
|          |        | Claims lists the names of resources, defined in <code>spec.resourceClaims</code> , that are used by this container.  |
| claims   | array  | <p>This is an alpha field and requires enabling the <code>DynamicResourceAllocation</code> feature gate.</p> <p>This field is immutable. It can only be set for containers.</p>  |
| limits   | object | <p>Limits describes the maximum amount of compute resources allowed. More info: <a href="https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/">https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/</a></p>  |
| requests | object | <p>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: <a href="https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/">https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/</a></p> |

## .spec.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.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.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.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.expose

## Description

Expose configures how Redis is exposed outside the Kubernetes cluster. Controls service types, node ports, and load balancer settings.

## Type

object

| Property                            | Type    | Description   |
|-------------------------------------|---------|---|
| <code>accessPort</code>             | integer | AccessPort defines the lb access nodeport   |
| <code>annotations</code>            | object  | The annotations of the service which attached to services   |
| <code>dataStorageNodePortMap</code> | object  | NodePortMap defines the map of the nodeport for redis sentinel only<br>Reversed for 3.14 backward compatibility |

| Property                                 | Type                 | Description   |
|--|----------------------|---|
| <code>dataStorageNodePortSequence</code> | <code>string</code>  | NodePortMap defines the map of the nodeport for redis nodes<br>NodePortSequence defines the sequence of the nodeport for redis cluster only |
| <code>enableNodePort</code>              | <code>boolean</code> | EnableNodePort defines if the nodeport is enabled   |
| <code>type</code>                        | <code>string</code>  | ServiceType defines the type of the all related service   |

## `.spec.expose.annotations`

### Description

The annotations of the service which attached to services

### Type

`object`

## `.spec.expose.dataStorageNodePortMap`

### Description

NodePortMap defines the map of the nodeport for redis sentinel only Reversed for 3.14 backward compatibility

### Type

`object`

## .spec.modules

### Description

Modules defines a list of Redis modules to be loaded into the Redis instance. Each module is specified by its name and version. Modules are loaded at startup and can extend Redis functionality.

### Type

array

## .spec.modules[]

### Description

RedisModule defines the module for Redis

### Type

object

### Required

path

| Property | Type   | Description  |
|----------|--------|--|
| args     | array  | Args args for module<br>Supported for redis 8.0+   |
| path     | string | Path of redis module.<br>.so suffix will be appended if path not suffixed provided if name is a full path, the full path will be used else the module will be loaded from /usr/local/lib/redis/modules |

## .spec.modules[].args

## Description

Args args for module Supported for redis 8.0+

## Type

array

## .spec.modules[].args[]

## Type

string

## .spec.nodeSelector

## Description

NodeSelector constrains Redis pods to nodes with matching labels. See: <https://kubernetes.io/docs/concepts/scheduling-eviction/assign-pod-node/#nodeselector>

## Type

object

## .spec.patches

## Description

Patches allows customization of generated Kubernetes resources. Enables fine-grained control over services and other child resources.

## Type

object

| Property | Type  | Description  |
|----------|-------|--|
| services | array | Services allows customizing the Kubernetes Services created for Redis. Can modify properties like port configuration, load balancer settings, and annotations. |

## .spec.patches.services

### Description

Services allows customizing the Kubernetes Services created for Redis. Can modify properties like port configuration, load balancer settings, and annotations.

### Type

array

## .spec.patches.services[]

### Description

Patch configuration for the Service created to serve traffic to the cluster. Allows for the manifest of the created Service to be overwritten with custom configuration.

### Type

object

| Property | Type         | Description  |
|----------|--------------|--|
| metadata | ObjectMeta ↗ | Metadata contains Service metadata like name, namespace, labels, and annotations.  |
| spec     | object       | Spec defines the Service behavior including type, ports, and selectors. See: <a href="https://kubernetes.io/docs/concepts/services-networking/service/">https://kubernetes.io/docs/concepts/services-networking/service/</a> ↗ |

## .spec.patches.services[].spec

### Description

Spec defines the Service behavior including type, ports, and selectors. See: <https://kubernetes.io/docs/concepts/services-networking/service/>

## Type

object

| Property                                   | Type    | Description  |
|--|---------|--|
| <code>allocateLoadBalancerNodePorts</code> | boolean | <p><code>allocateLoadBalancerNodePorts</code> defines if NodePorts will be automatically allocated for services with type <code>LoadBalancer</code>. Default is "true". It may be set to "false" if the cluster load-balancer does not rely on NodePorts. If the caller requests specific NodePorts (by specifying a value), those requests will be respected, regardless of this field. This field may only be set for services with type <code>LoadBalancer</code> and will be cleared if the type is changed to any other type.</p>   |
| <code>clusterIP</code>                     | string  | <p><code>clusterIP</code> is the IP address of the service and is usually assigned randomly. If an address is specified manually, it must be in-range (as per system configuration), and is not in use, it will be allocated to the service; otherwise creation of the service will fail. This field must not be changed through updates unless the type field is also being changed to <code>ExternalName</code> (which requires this field to be blank) or the type field is being changed from <code>ExternalName</code> (in which case this field may optionally be specified, as described above). Valid values are "None", empty string (""), or a valid IP address. Setting this to "None" makes a "headless service" (no virtual IP), which is useful when direct endpoint connections are preferred and proxying is not required. Only applies to</p> |

| Property                       | Type                      | Description  |
|--------------------------------|---------------------------|--|
|                                |                           | <p>types ClusterIP, NodePort, and LoadBalancer. If this field is specified when creating a Service of type ExternalName, creation will fail. This field will be wiped when updating a Service to type ExternalName.</p> <p>More info:</p> <p><a href="https://kubernetes.io/docs/concepts/service-networking/service/#virtual-ips-and-service-proxies">https://kubernetes.io/docs/concepts/service-networking/service/#virtual-ips-and-service-proxies</a></p>   |
| <p><code>clusterIPs</code></p> | <p><code>array</code></p> | <p>ClusterIPs is a list of IP addresses assigned to this service, and are usually assigned randomly. If an address is specified manually, is in-range (as per system configuration), and is not in use, it will be allocated to the service; otherwise creation of the service will fail. This field may not be changed through updates unless the type field is also being changed to ExternalName (which requires this field to be empty) or the type field is being changed from ExternalName (in which case this field may optionally be specified, as described above). Valid values are "None", empty string (""), or a valid IP address. Setting this to "None" makes a "headless service" (no virtual IP), which is useful when direct endpoint connections are preferred and proxying is not required. Only applies to types ClusterIP, NodePort, and LoadBalancer. If this field is specified when creating a Service of type ExternalName, creation will fail. This field will be wiped when updating a Service to type ExternalName. If this field is not specified,</p> |

| Property     | Type   | Description   |
|--------------|--------|---|
|              |        | <p>will be initialized from the clusterIP field. If this field is specified, clients must ensure that clusterIPs[0] and clusterIP have the same value.</p> <p>This field may hold a maximum of two entries (dual-stack IPs, in either order). These IPs must correspond to the values c the ipFamilies field. Both clusterIPs and ipFamilies are governed by the ipFamilyPolicy field. More info: <a href="https://kubernetes.io/docs/concepts/service-networking/service/#virtual-ips-and-service-proxies">https://kubernetes.io/docs/concepts/service-networking/service/#virtual-ips-and-service-proxies</a> ↗</p> |
| externalIPs  | array  | <p>externalIPs is a list of IP addresses for whi nodes in the cluster will also accept traffic i this service. These IPs are not managed b Kubernetes. The user is responsible for ensuring that traffic arrives at a node with this IP. A common example is external load balancers that are not part of the Kubernetes system.</p>  |
| externalName | string | <p>externalName is the external reference tha discovery mechanisms will return as an ali for this service (e.g. a DNS CNAME record No proxying will be involved. Must be a lowercase RFC-1123 hostname (<a href="https://tools.ietf.org/html/rfc1123">https://tools.ietf.org/html/rfc1123</a> ↗) and requires <code>type</code> to be "ExternalName".</p>   |

| Property                           | Type                 | Description  |
|------------------------------------|----------------------|--|
| <code>externalTrafficPolicy</code> | <code>string</code>  | <p><code>externalTrafficPolicy</code> describes how nodes distribute service traffic they receive on one of the Service's "externally-facing" addresses (NodePorts, ExternalIPs, and LoadBalancer IPs). If set to "Local", the proxy will configure the service in a way that assumes that external load balancers will take care of balancing the service traffic between nodes, and so each node will deliver traffic only to the node-local endpoints of the service, without masquerading the client source IP. (Traffic mistakenly sent to a node with no endpoint will be dropped.) The default value, "Cluster", uses the standard behavior of routing to all endpoints evenly (possibly modified by topology and other features). Note that traffic sent to an External IP or LoadBalancer IP from within the cluster will always get "Cluster" semantics, but clients sending to a NodePort from within the cluster may need to take traffic policy into account when picking a node.</p> |
| <code>healthCheckNodePort</code>   | <code>integer</code> | <p><code>healthCheckNodePort</code> specifies the healthcheck nodePort for the service. This only applies when type is set to LoadBalancer and <code>externalTrafficPolicy</code> is set to Local. If a value is specified, it is in-range, and is not in use, it will be used. If not specified, a value will be automatically allocated. External systems (e.g. load-balancers) can use this port to determine if</p>  |

| Property                           | Type                | Description   |
|------------------------------------|---------------------|---|
|                                    |                     | given node holds endpoints for this service or not. If this field is specified when creating a Service which does not need it, creation will fail. This field will be wiped when updating a Service to no longer need it (e.g. changing type). This field cannot be updated once set.   |
| <code>internalTrafficPolicy</code> | <code>string</code> | InternalTrafficPolicy describes how nodes distribute service traffic they receive on the ClusterIP. If set to "Local", the proxy will assume that pods only want to talk to endpoints of the service on the same node as the pod, dropping the traffic if there are local endpoints. The default value, "Cluster", uses the standard behavior of routing to all endpoints evenly (possibly modified by topology and other features).  |
| <code>ipFamilies</code>            | <code>array</code>  | IPFamilies is a list of IP families (e.g. IPv4, IPv6) assigned to this service. This field is usually assigned automatically based on cluster configuration and the ipFamilyPolicy field. If this field is specified manually, the requested family is available in the cluster, and ipFamilyPolicy allows it, it will be used otherwise creation of the service will fail. This field is conditionally mutable: it allows for adding or removing a secondary IP family, but it does not allow changing the primary IP family of the Service. Valid values are "IPv4" and "IPv6". This field only applies |

| Property          | Type   | Description  |
|-------------------|--------|--|
|                   |        | <p>to Services of types ClusterIP, NodePort, and LoadBalancer, and does apply to "headless" services. This field will be wiped when updating a Service to type ExternalName.</p> <p>This field may hold a maximum of two entries (dual-stack families, in either order) These families must correspond to the values of the clusterIPs field, if specified. Both clusterIPs and ipFamilies are governed by the ipFamilyPolicy field.</p>   |
| ipFamilyPolicy    | string | <p>IPFamilyPolicy represents the dual-stack-ness requested or required by this Service there is no value provided, then this field will be set to SingleStack. Services can be "SingleStack" (a single IP family), "PreferDualStack" (two IP families on dual-stack configured clusters or a single IP family on single-stack clusters), or "RequireDualStack" (two IP families on dual stack configured clusters, otherwise fail). The ipFamilies and clusterIPs fields depend on the value of this field. This field will be wiped when updating a service to type ExternalName.</p> |
| loadBalancerClass | string | <p>loadBalancerClass is the class of the load balancer implementation this Service belongs to. If specified, the value of this field must be a label-style identifier, with an optional prefix, e.g. "internal-vip" or</p>   |

| Property                    | Type                | Description  |
|-----------------------------|---------------------|--|
|                             |                     | <p>"example.com/internal-vip". Unprefixed names are reserved for end-users. This field can only be set when the Service type is 'LoadBalancer'. If not set, the default load balancer implementation is used, today this is typically done through the cloud provider integration, but should apply for any default implementation. If set, it is assumed that a load balancer implementation is watching for Services with a matching class. Any default load balancer implementation (e.g. cloud providers) should ignore Services that set this field. This field can only be set when creating or updating a Service to type 'LoadBalancer'. Once set, it can not be changed. This field will be wiped when a service is updated to a non 'LoadBalancer' type.</p> |
| <code>loadBalancerIP</code> | <code>string</code> | <p>Only applies to Service Type: LoadBalancer. This feature depends on whether the underlying cloud-provider supports specifying the loadBalancerIP when a load balancer is created. This field will be ignored if the cloud-provider does not support the feature. Deprecated: This field was underspecified and its meaning varies across implementations. Using it is non-portable and it may not support dual-stack. Users are encouraged to use implementation-specific annotations when available.</p>   |

| Property                              | Type                 | Description   |
|---------------------------------------|----------------------|---|
| <code>loadBalancerSourceRanges</code> | <code>array</code>   | <p>If specified and supported by the platform, this will restrict traffic through the cloud-provider load-balancer will be restricted to the specified client IPs. This field will be ignored if the cloud-provider does not support the feature." More info: <a href="https://kubernetes.io/docs/tasks/access-application-cluster/create-external-load-balancer/">https://kubernetes.io/docs/tasks/access-application-cluster/create-external-load-balancer/</a> ↗</p>   |
| <code>ports</code>                    | <code>array</code>   | <p>The list of ports that are exposed by this service. More info: <a href="https://kubernetes.io/docs/concepts/service-networking/service/#virtual-ips-and-service-proxies">https://kubernetes.io/docs/concepts/service-networking/service/#virtual-ips-and-service-proxies</a> ↗</p>   |
| <code>publishNotReadyAddresses</code> | <code>boolean</code> | <p><code>publishNotReadyAddresses</code> indicates that any agent which deals with endpoints for this Service should disregard any indications of ready/not-ready. The primary use case for setting this field is for a StatefulSet's Headless Service to propagate SRV DNS records for its Pods for the purpose of peer discovery. The Kubernetes controllers that generate Endpoints and EndpointSlice resources for Services interpret this to mean that all endpoints are considered "ready" even if the Pods themselves are not. Agents which consume only Kubernetes generated endpoints through the Endpoints or</p> |

| Property                           | Type                | Description   |
|------------------------------------|---------------------|---|
|                                    |                     | EndpointSlice resources can safely assume this behavior.  |
| <code>selector</code>              | <code>object</code> | Route service traffic to pods with label key: and values matching this selector. If empty or not present, the service is assumed to have an external process managing its endpoints, which Kubernetes will not modify. Only applies to types ClusterIP, NodePort, and LoadBalancer. Ignored if type is ExternalName. More info: <a href="https://kubernetes.io/docs/concepts/service-networking/service/">https://kubernetes.io/docs/concepts/service-networking/service/</a> |
| <code>sessionAffinity</code>       | <code>string</code> | Supports "ClientIP" and "None". Used to maintain session affinity. Enable client IP based session affinity. Must be ClientIP or None. Defaults to None. More info: <a href="https://kubernetes.io/docs/concepts/service-networking/service/#virtual-ips-and-service-proxies">https://kubernetes.io/docs/concepts/service-networking/service/#virtual-ips-and-service-proxies</a>  |
| <code>sessionAffinityConfig</code> | <code>object</code> | <code>sessionAffinityConfig</code> contains the configurations of session affinity.   |
| <code>trafficDistribution</code>   | <code>string</code> | <code>TrafficDistribution</code> offers a way to express preferences for how traffic is distributed to Service endpoints. Implementations can use this field as a hint, but are not required to guarantee strict adherence. If the field is not   |

| Property    | Type          | Description  |
|-------------|---------------|--|
|             |               | <p>set, the implementation will apply its default routing strategy. If set to "PreferClose", implementations should prioritize endpoints that are in the same zone.</p>  |
| <p>type</p> | <p>string</p> | <p>type determines how the Service is exposed. Defaults to ClusterIP. Valid options are ExternalName, ClusterIP, NodePort, and LoadBalancer. "ClusterIP" allocates a cluster-internal IP address for load-balancing to endpoints. Endpoints are determined by the selector or if that is not specified, by manual construction of an Endpoints object or EndpointSlice objects. If clusterIP is "None", no virtual IP is allocated and the endpoints are published as a set of endpoints rather than a virtual IP. "NodePort" builds on ClusterIP and allocates a port on every node which routes to the same endpoints as the clusterIP. "LoadBalancer" builds on NodePort and creates an external load-balancer (if supported in the current cloud) which routes to the same endpoints as the clusterIP. "ExternalName" aliases this service to the specified externalName. Several other fields do not apply to ExternalName services. More info: <a href="https://kubernetes.io/docs/concepts/services-networking/service/#publishing-services-service-types">https://kubernetes.io/docs/concepts/services-networking/service/#publishing-services-service-types</a></p> |

**.spec.patches.services[].spec.clusterIPs**

## Description

ClusterIPs is a list of IP addresses assigned to this service, and are usually assigned randomly. If an address is specified manually, is in-range (as per system configuration), and is not in use, it will be allocated to the service; otherwise creation of the service will fail. This field may not be changed through updates unless the type field is also being changed to ExternalName (which requires this field to be empty) or the type field is being changed from ExternalName (in which case this field may optionally be specified, as describe above). Valid values are "None", empty string (""), or a valid IP address. Setting this to "None" makes a "headless service" (no virtual IP), which is useful when direct endpoint connections are preferred and proxying is not required. Only applies to types ClusterIP, NodePort, and LoadBalancer. If this field is specified when creating a Service of type ExternalName, creation will fail. This field will be wiped when updating a Service to type ExternalName. If this field is not specified, it will be initialized from the clusterIP field. If this field is specified, clients must ensure that clusterIPs[0] and clusterIP have the same value. This field may hold a maximum of two entries (dual-stack IPs, in either order). These IPs must correspond to the values of the ipFamilies field. Both clusterIPs and ipFamilies are governed by the ipFamilyPolicy field. More info: <https://kubernetes.io/docs/concepts/services-networking/service/#virtual-ips-and-service-proxies>

## Type

array

## `.spec.patches.services[].spec.clusterIPs[]`

## Type

string

## `.spec.patches.services[].spec.externalIPs`

## Description

externalIPs is a list of IP addresses for which nodes in the cluster will also accept traffic for this service. These IPs are not managed by Kubernetes. The user is responsible for ensuring that traffic arrives at a node with this IP. A common example is external load-balancers that are not part of the Kubernetes system.

## Type

array

## `.spec.patches.services[].spec.externalIPs[]`

### Type

string

## `.spec.patches.services[].spec.ipFamilies`

### Description

IPFamilies is a list of IP families (e.g. IPv4, IPv6) assigned to this service. This field is usually assigned automatically based on cluster configuration and the ipFamilyPolicy field. If this field is specified manually, the requested family is available in the cluster, and ipFamilyPolicy allows it, it will be used; otherwise creation of the service will fail. This field is conditionally mutable: it allows for adding or removing a secondary IP family, but it does not allow changing the primary IP family of the Service. Valid values are "IPv4" and "IPv6". This field only applies to Services of types ClusterIP, NodePort, and LoadBalancer, and does not apply to "headless" services. This field will be wiped when updating a Service to type ExternalName. This field may hold a maximum of two entries (dual-stack families, in either order). These families must correspond to the values of the clusterIPs field, if specified. Both clusterIPs and ipFamilies are governed by the ipFamilyPolicy field.

### Type

array

## `.spec.patches.services[].spec.ipFamilies[]`

### Description

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).

### Type

string

## `.spec.patches.services[].spec.loadBalancerSourceRanges`

### Description

If specified and supported by the platform, this will restrict traffic through the cloud-provider load-balancer will be restricted to the specified client IPs. This field will be ignored if the cloud-provider does not support the feature." More info:

<https://kubernetes.io/docs/tasks/access-application-cluster/create-external-load-balancer/>

#### Type

array

## **.spec.patches.services[].spec.loadBalancerSourceRanges**

[]

#### Type

string

## **.spec.patches.services[].spec.ports**

#### Description

The list of ports that are exposed by this service. More info:

<https://kubernetes.io/docs/concepts/services-networking/service/#virtual-ips-and-service-proxies>

#### Type

array

## **.spec.patches.services[].spec.ports[]**

#### Description

ServicePort contains information on service's port.

#### Type

object

#### Required

port

| Property                 | Type                | Description  |
|--------------------------|---------------------|--|
| <code>appProtocol</code> | <code>string</code> | <p>The application protocol for this port. This is used as a hint for implementations to offer richer behavior for protocols that they understand. This field follows standard Kubernetes label syntax. Valid values are either:</p> <ul style="list-style-type: none"><li>• Un-prefixed protocol names - reserved for IANA standard service names (as per RFC-6335 and <a href="https://www.iana.org/assignments/service-names">https://www.iana.org/assignments/service-names</a> ).</li><li>• Kubernetes-defined prefixed names:<ul style="list-style-type: none"><li>• 'kubernetes.io/h2c' - HTTP/2 prior knowledge over cleartext as described in <a href="https://www.rfc-editor.org/rfc/rfc9113.html#name-starting-http-2-with-prior-">https://www.rfc-editor.org/rfc/rfc9113.html#name-starting-http-2-with-prior-</a></li><li>• 'kubernetes.io/ws' - WebSocket over cleartext as described in <a href="https://www.rfc-editor.org/rfc/rfc6455">https://www.rfc-editor.org/rfc/rfc6455</a></li><li>• 'kubernetes.io/wss' - WebSocket over TLS as described in <a href="https://www.rfc-editor.org/rfc/rfc6455">https://www.rfc-editor.org/rfc/rfc6455</a></li></ul></li><li>• Other protocols should use implementation-defined prefixed names such as mycompany.com/my-custom-protocol.</li></ul> |
| <code>name</code>        | <code>string</code> | <p>The name of this port within the service. This must be a DNS_LABEL. All ports within a ServiceSpec must have unique names. When considering the endpoints for a Service, this must match the 'name' field in the EndpointPort. Optional if only one ServicePort is defined on this service.</p>   |

| Property                | Type                 | Description  |
|-------------------------|----------------------|--|
| <code>nodePort</code>   | <code>integer</code> | <p>The port on each node on which this service is exposed when type is NodePort or LoadBalancer. Usually assigned by the system. If a value is specified, in-range, and not in use it will be used, otherwise the operation will fail. If not specified, a port will be allocated if this Service requires one. If this field is specified when creating a Service which does not need it, creation will fail. This field will be wiped when updating a Service to no longer need it (e.g. changing type from NodePort to ClusterIP). More info: <a href="https://kubernetes.io/docs/concepts/services-networking/service/#type-nodeport">https://kubernetes.io/docs/concepts/services-networking/service/#type-nodeport</a></p> |
| <code>port</code>       | <code>integer</code> | <p>The port that will be exposed by this service.</p>  |
| <code>protocol</code>   | <code>string</code>  | <p>The IP protocol for this port. Supports "TCP", "UDP", and "SCTP". Default is TCP.</p>   |
| <code>targetPort</code> |                      | <p>Number or name of the port to access on the pods targeted by the service. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME. If this is a string, it will be looked up as a named port in the target Pod's container ports. If this is not specified, the value of the 'port' field is used (an identity map). This field is ignored for services with clusterIP=None, and should be omitted or set equal to the 'port' field. More info: <a href="https://kubernetes.io/docs/concepts/services-networking/service/#defining-a-service">https://kubernetes.io/docs/concepts/services-networking/service/#defining-a-service</a></p>   |

## `.spec.patches.services[].spec.selector`

### Description

Route service traffic to pods with label keys and values matching this selector. If empty or not present, the service is assumed to have an external process managing its endpoints, which Kubernetes will not modify. Only applies to types ClusterIP, NodePort, and LoadBalancer. Ignored if type is ExternalName. More info: <https://kubernetes.io/docs/concepts/services-networking/service/>

### Type

object

## `.spec.patches.services[].spec.sessionAffinityConfig`

### Description

sessionAffinityConfig contains the configurations of session affinity.

### Type

object

| Property              | Type   | Description   |
|-----------------------|--------|---|
| <code>clientIP</code> | object | clientIP contains the configurations of Client IP based session affinity. |

## `.spec.patches.services[].spec.sessionAffinityConfig.clientIP`

### Description

clientIP contains the configurations of Client IP based session affinity.

### Type

object

| Property                    | Type                 | Description  |
|-----------------------------|----------------------|--|
| <code>timeoutSeconds</code> | <code>integer</code> | timeoutSeconds specifies the seconds of ClientIP type session sticky time. The value must be >0 && <=86400(for 1 day) if ServiceAffinity == "ClientIP". Default value is 10800(for 3 hours). |

## .spec.persistent

### Description

Persistent configures persistent storage for Redis data. If specified, a PersistentVolumeClaim will be created for each Redis node.

### Type

`object`

### Required

`storageClassName`

| Property                      | Type                | Description  |
|-------------------------------|---------------------|--|
| <code>storageClassName</code> | <code>string</code> | StorageClassName specifies the name of the Kubernetes StorageClass to use for Redis persistent volumes. The storage class determines provisioner, reclaim policy, and mount options. |

## .spec.podAnnotations

### Description

PodAnnotations are custom annotations to add to all Redis pods. Useful for integrations with service meshes, monitoring tools, etc.

### Type

object

## .spec.redisProxy

### Description

RedisProxy configures an optional proxy layer in front of Redis. Deprecated: Will be removed in 3.20.

### Type

object

| Property                  | Type    | Description   |
|---------------------------|---------|---|
| <code>affinity</code>     | object  | Affinity defines pod scheduling constraints for proxy pods. Controls pod placement based on node affinity, pod affinity, and anti-affinity. |
| <code>config</code>       | object  | Config provides additional configuration parameters for the Redis Proxy. Key-value pairs corresponding to proxy configuration options.      |
| <code>enable</code>       | boolean | Enable toggles whether the Redis Proxy service is deployed. When true, a proxy layer will be created between clients and Redis.             |
| <code>image</code>        | string  | Image specifies the container image used for the Redis Proxy. Format: <code>[registry]/[repository]:[tag]</code>                            |
| <code>nodeSelector</code> | object  | NodeSelector constrains proxy pods to nodes with matching labels. See:  |

| Property                 | Type                 | Description  |
|--------------------------|----------------------|--|
|                          |                      | <a href="https://kubernetes.io/docs/concepts/scheduling-eviction/assign-pod-node/#nodeselector">https://kubernetes.io/docs/concepts/scheduling-eviction/assign-pod-node/#nodeselector</a> ↗  |
| <code>replicas</code>    | <code>integer</code> | Replicas defines how many proxy instances to deploy for load balancing. Higher values provide better availability and throughput.  |
| <code>resources</code>   | <code>object</code>  | Resources defines compute resource requirements for the proxy containers. Controls CPU and memory limits/requests.   |
| <code>tolerations</code> | <code>array</code>   | Tolerations allow the proxy pods to be scheduled on nodes with matching taints. See: <a href="https://kubernetes.io/docs/concepts/scheduling-eviction/taint-and-toleration/">https://kubernetes.io/docs/concepts/scheduling-eviction/taint-and-toleration/</a> ↗ |

## .spec.redisProxy.affinity

### Description

Affinity defines pod scheduling constraints for proxy pods. Controls pod placement based on node affinity, pod affinity, and anti-affinity.

### 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.redisProxy.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 o |

| Property   | Type                       | Description   |
|--|----------------------------|---|
|  |                            | <p>scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution affinity expressions, etc.) to compute a sum by iterating through the elements of this field adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest sum are the most preferred.</p>                           |
| <p><code>requiredDuringSchedulingIgnoredDuringExecution</code></p> | <p><code>object</code></p> | <p>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 system restart or update), the system may or may not try to eventually evict the pod from its node.</p> |

## .spec.redisProxy.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.redisProxy.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. |

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

## `.spec.redisProxy.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.redisProxy.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions`

### Description

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

### Type

`array`

# `.spec.redisProxy.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.redisProxy.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpres`

## sions[].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.redisProxy.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values[]

### Type

string

## .spec.redisProxy.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields

### Description

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

### Type

array

## .spec.redisProxy.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.redisProxy.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.redisProxy.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values[]`

### Type

string

## `.spec.redisProxy.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> | array | Required. A list of node selector terms. The terms are ORed. |

## `.spec.redisProxy.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms`

## Description

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

## Type

array

# `.spec.redisProxy.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.redisProxy.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions`

## Description

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

## Type

array

# `.spec.redisProxy.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.  |
| 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.redisProxy.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[]`

## hExpressions[].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.redisProxy.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values[]

### Type

string

## .spec.redisProxy.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields

### Description

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

### Type

array

## .spec.redisProxy.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.redisProxy.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.redisProxy.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values[]**

## Type

string

**.spec.redisProxy.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, 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 o |

| Property   | Type                      | Description  |
|--|---------------------------|--|
|  |                           | <p>scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution, affinity expressions, etc.) to compute a sum by iterating through the elements of this field adding "weight" to the sum if the node has a label which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.</p>  |
| <p><code>requiredDuringSchedulingIgnoredDuringExecution</code></p> | <p><code>array</code></p> | <p>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</p> |

| Property | Type | Description                                    |
|----------|------|--|
|          |      | intersected, i.e. all terms must be satisfied. |

## `.spec.redisProxy.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.redisProxy.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.redisProxy.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 <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 <code>labelSelector</code> as <code>key in (value)</code> to select |

| Property                              | Type          | Description   |
|---------------------------------------|---------------|---|
|                                       |               | <p>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.</p>   |
| <p><code>mismatchLabelKeys</code></p> | <p>array</p>  | <p><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.</p> |
| <p><code>namespaceSelector</code></p> | <p>object</p> | <p>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.</p>  |
|                                       |               |   |

| 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.redisProxy.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>  | <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.redisProxy.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions`

### Description

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

### Type

`array`

## `.spec.redisProxy.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.redisProxy.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.redisProxy.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSele**

## ctor.matchExpressions[].values[]

### Type

string

## .spec.redisProxy.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.redisProxy.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.

### Type

array

## `.spec.redisProxy.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]`

### Type

string

## `.spec.redisProxy.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.

### Type

array

## `.spec.redisProxy.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]`

### Type

string

## `.spec.redisProxy.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespa`

# ceSelector

## 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.redisProxy.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespacesSelector.matchExpressions

## Description

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

## Type

array

# `.spec.redisProxy.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.   |
| 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.redisProxy.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.redisProxy.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]**

#### Type

string

**.spec.redisProxy.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.redisProxy.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.redisProxy.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]**

### Type

string

## **.spec.redisProxy.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.redisProxy.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>     | 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. |
| <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 not in</code>  |

| Property                       | Type                | Description  |
|--------------------------------|---------------------|--|
|                                |                     | <p><code>(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.</p> |
| <code>namespaceSelector</code> | <code>object</code> | <p>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.</p>  |
| <code>namespaces</code>        | <code>array</code>  | <p><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".</p>   |
|                                |                     |  |

| 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.redisProxy.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>  | <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.redisProxy.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

### Description

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

### Type

array

## `.spec.redisProxy.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. |
|          |        |  |

| 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.redisProxy.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.redisProxy.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]**

### Type

string

## **.spec.redisProxy.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.redisProxy.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.

### Type

array

## **.spec.redisProxy.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]**

### Type

string

## **.spec.redisProxy.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.

### Type

array

## `.spec.redisProxy.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]`

### Type

string

## `.spec.redisProxy.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 |

| Property | Type | Description  |
|----------|------|--|
|          |      | "key", the operator is "In", and the values array contains only "value". The requirements are ANDed. |

## `.spec.redisProxy.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions`

### Description

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

### Type

array

## `.spec.redisProxy.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. |

| Property              | Type                | Description  |
|-----------------------|---------------------|--|
| <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.redisProxy.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.redisProxy.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]`

### Type

`string`

## **.spec.redisProxy.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.redisProxy.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.redisProxy.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]**

### **Type**

string

## **.spec.redisProxy.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> | <p>The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field. 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 a podAffinityTerm which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.</p> |
| <code>requiredDuringSchedulingIgnoredDuringExecution</code>  | <code>array</code> | <p>If the anti-affinity requirements specified in this field are not met by a node, the pod will not be scheduled onto the node.</p>  |

| Property | Type | Description   |
|----------|------|---|
|          |      | will not be scheduled 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 pod label update), the system may or may not try to eventually evict pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm intersected, i.e. all terms must be satisfied. |

## `.spec.redisProxy.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.redisProxy.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  |
|-----------------|---------|--|
| 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.redisProxy.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 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. |

| Property                       | Type                | Description  |
|--------------------------------|---------------------|--|
| <code>mismatchLabelKeys</code> | <code>array</code>  | <p>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 <code>mismatchLabelKeys</code> and <code>labelSelector</code>. Also, <code>mismatchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set.</p> |
| <code>namespaceSelector</code> | <code>object</code> | <p>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.</p>   |
| <code>namespaces</code>        | <code>array</code>  | <p>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 <code>namespaceSelector</code>. null or empty namespaces list and null <code>namespaceSelector</code> means "this pod's namespace".</p>  |

| 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.redisProxy.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.redisProxy.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions`

### Description

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

### Type

array

## `.spec.redisProxy.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. |
|          |        |  |

| 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.redisProxy.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.redisProxy.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]**

### Type

string

## **.spec.redisProxy.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.redisProxy.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.

### Type

array

## **.spec.redisProxy.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]**

### Type

string

## **.spec.redisProxy.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.

## Type

array

## `.spec.redisProxy.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]`

## Type

string

## `.spec.redisProxy.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> | <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.redisProxy.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces.selector.matchExpressions`

### Description

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

### Type

`array`

## `.spec.redisProxy.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.redisProxy.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.redisProxy.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces.selector.matchExpressions[].values**

## namespaceSelector.matchExpressions[].values[]

### Type

string

## .spec.redisProxy.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

### 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.redisProxy.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.redisProxy.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

## Type

string

# .spec.redisProxy.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.redisProxy.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 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 <code>matchLabelKeys</code> and <code>labelSelector</code> . Also, <code>matchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. |

| Property                       | Type                | Description  |
|--------------------------------|---------------------|--|
| <code>mismatchLabelKeys</code> | <code>array</code>  | <p>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 <code>mismatchLabelKeys</code> and <code>labelSelector</code>. Also, <code>mismatchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set.</p> |
| <code>namespaceSelector</code> | <code>object</code> | <p>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.</p>   |
| <code>namespaces</code>        | <code>array</code>  | <p><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".</p>   |

| 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.redisProxy.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> | <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.redisProxy.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

### Description

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

### Type

array

## `.spec.redisProxy.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. |
|          |        |  |

| 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.redisProxy.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.redisProxy.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]**

### **Type**

string

## **.spec.redisProxy.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.redisProxy.affinity.podAntiAffinity.requiredDuringScheduleduningIgnoredDuringExecution[].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.

### Type

array

## **.spec.redisProxy.affinity.podAntiAffinity.requiredDuringScheduleduningIgnoredDuringExecution[].matchLabelKeys[]**

### Type

string

## **.spec.redisProxy.affinity.podAntiAffinity.requiredDuringScheduleduningIgnoredDuringExecution[].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.

### Type

array

## `.spec.redisProxy.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]`

### Type

string

## `.spec.redisProxy.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 |

| Property | Type | Description  |
|----------|------|--|
|          |      | "key", the operator is "In", and the values array contains only "value". The requirements are ANDed. |

## `.spec.redisProxy.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions`

### Description

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

### Type

array

## `.spec.redisProxy.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. |

| Property              | Type                | Description  |
|-----------------------|---------------------|--|
| <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.redisProxy.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.redisProxy.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]`

### Type

`string`

## `.spec.redisProxy.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.redisProxy.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.redisProxy.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]`

### Type

string

## `.spec.redisProxy.config`

### Description

Config provides additional configuration parameters for the Redis Proxy. Key-value pairs corresponding to proxy configuration options.

### Type

object

## .spec.redisProxy.nodeSelector

### Description

NodeSelector constrains proxy pods to nodes with matching labels. See: <https://kubernetes.io/docs/concepts/scheduling-eviction/assign-pod-node/#nodeselector>

### Type

object

## .spec.redisProxy.resources

### Description

Resources defines compute resource requirements for the proxy containers. Controls CPU and memory limits/requests.

### Type

object

| Property | Type   | Description   |
|----------|--------|---|
|          |        | Claims lists the names of resources, defined in <code>spec.resourceClaims</code> , that are used by this container.   |
| claims   | array  | <p>This is an alpha field and requires enabling the <code>DynamicResourceAllocation</code> feature gate.</p> <p>This field is immutable. It can only be set for containers.</p> |
| limits   | object | Limits describes the maximum amount of compute resources allowed. More info:  |

| Property              | Type                | Description   |
|-----------------------|---------------------|---|
|                       |                     | <a href="https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/">https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/</a> ↗   |
| <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: <a href="https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/">https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/</a> ↗ |

## `.spec.redisProxy.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.redisProxy.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.redisProxy.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.redisProxy.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.redisProxy.tolerations

### Description

Tolerations allow the proxy pods to be scheduled on nodes with matching taints. See: <https://kubernetes.io/docs/concepts/scheduling-eviction/taint-and-toleration/>

## Type

array

## .spec.redisProxy.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. |

| Property                       | Type                 | Description   |
|--------------------------------|----------------------|---|
| <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.replicas

### Description

Replicas defines the number of Redis nodes for different architectures. Configuration varies based on the selected architecture (cluster or sentinel).

### Type

`object`

| Property              | Type                | Description  |
|-----------------------|---------------------|--|
| <code>cluster</code>  | <code>object</code> | Cluster holds replica settings specific to Redis Cluster architecture. Configures shard count and replica count per shard. |
| <code>sentinel</code> | <code>object</code> | Sentinel holds replica settings specific to Redis Sentinel architecture. Configures master and replica count.              |

## .spec.replicas.cluster

### Description

Cluster holds replica settings specific to Redis Cluster architecture. Configures shard count and replica count per shard.

### Type

object

### Required

shard

| Property | Type    | Description   |
|----------|---------|---|
| shard    | integer | Shard specifies the number of master nodes in the Redis Cluster. Each shard handles a subset of the hash slots (16384 total slots). Minimum value: 3                  |
| shards   | array   | Shards allows custom configuration of hash slot ranges for each shard. This configuration only works for newly created instances and cannot be updated after startup. |
| slave    | integer | Slave specifies the number of replica nodes per master in the Redis Cluster. Each replica maintains a copy of its master's data for high availability. Range: 0-5     |

## .spec.replicas.cluster.shards

### Description

Shards allows custom configuration of hash slot ranges for each shard. This configuration only works for newly created instances and cannot be updated after startup.

### Type

`array`

## `.spec.replicas.cluster.shards[]`

### Type

`object`

| Property           | Type                | Description  |
|--------------------|---------------------|--|
| <code>slots</code> | <code>string</code> | Slots is the slot range for the shard, eg: 0-1000,1002,1005-1100 |

## `.spec.replicas.sentinel`

### Description

Sentinel holds replica settings specific to Redis Sentinel architecture. Configures master and replica count.

### Type

`object`

### Required

`master`

| Property            | Type                 | Description   |
|---------------------|----------------------|---|
| <code>master</code> | <code>integer</code> | Master specifies the number of Redis master nodes. For Sentinel architecture, only 1 master is supported. |
| <code>slave</code>  | <code>integer</code> | Slave specifies the number of Redis replica nodes that replicate from the master. Range: 0-5              |

# .spec.resources

## Description

Resources defines the compute resource requirements for Redis containers. This includes CPU and memory limits/requests.

## Type

object

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

## .spec.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.resources.claims[]

### Description

ResourceClaim references one entry in `PodSpec.ResourceClaims`.

### Type

object

### Required

name

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

### Description

Restore configures data restoration from a backup. Specifies the backup source and restoration parameters.

## Type

object

| Property        | Type   | Description   |
|-----------------|--------|---|
| backupName      | string |   |
| image           | string |   |
| imagePullPolicy | string | PullPolicy describes a policy for if/when to pull a container image |

# .spec.securityContext

## Description

SecurityContext defines security settings for Redis pods, including user/group IDs, filesystem permissions, and SELinux context.

## 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 | <p>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:</p> <ol style="list-style-type: none"> <li>1. The owning GID will be the FSGroup</li> <li>2. The setgid bit is set (new files created in the volume will be owned by FSGroup)</li> <li>3. The permission bits are OR'd with rw-rw- ---</li> </ol> <p>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.</p> |

| Property                         | Type                 | Description  |
|----------------------------------|----------------------|--|
| <code>fsGroupChangePolicy</code> | <code>string</code>  | <p><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 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 <code>spec.os.name</code> is windows.</p> |
| <code>runAsGroup</code>          | <code>integer</code> | <p>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 windows.</p>   |
| <code>runAsNonRoot</code>        | <code>boolean</code> | <p>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</p>  |

| Property            | Type    | Description   |
|---------------------|---------|---|
|                     |         | PodSecurityContext, the value specified in SecurityContext takes precedence.  |
| runAsUser           | integer | <p>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.</p>   |
| seLinuxChangePolicy | string  | <p>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".</p> <p>"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.</p> <p>"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.</p> |

| Property                           | Type                       | Description  |
|------------------------------------|----------------------------|--|
|                                    |                            | <p>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 <code>CSIDriver</code> instance. Other volumes are always re-labelled recursively.</p> <p>"MountOption" value is allowed only when SELinuxMount feature gate is enabled.</p> <p>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.</p> <p>This field affects only Pods that have SELinux label set, either in <code>PodSecurityContext</code> or in <code>SecurityContext</code> of all containers.</p> <p>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>.</p> |
| <p><code>seLinuxOptions</code></p> | <p><code>object</code></p> | <p>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 <code>SecurityContext</code>. If set in both <code>SecurityContext</code> and <code>PodSecurityContext</code>, the value specified in <code>SecurityContext</code> takes</p>   |

| Property                              | Type                | Description   |
|---------------------------------------|---------------------|---|
|                                       |                     | 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.  |
| <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  |

| Property                    | Type                | Description  |
|-----------------------------|---------------------|--|
|                             |                     | 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.   |
| <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.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 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.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. |

| 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.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 <code>type</code> is "Localhost". Must NOT be set for any other type. |

| Property | Type   | Description  |
|----------|--------|--|
|          |        | <p>type indicates which kind of seccomp profile will be applied. Valid options are:</p>  |
| type     | string | <p>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.</p> |

## `.spec.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.securityContext.supplementalGroups[]`

### Type

integer

## `.spec.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.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.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  |
|-------------------------------------|----------------------|--|
| <code>gmsaCredentialSpec</code>     | <code>string</code>  | GMSACredentialSpec is where the GMSA admission webhook ( <a href="https://github.com/kubernetes-sigs/windows-gmsa">https://github.com/kubernetes-sigs/windows-gmsa</a> ) 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.sentinel

## Description

Sentinel configures Redis Sentinel for high availability. Only applicable when arch is set to "sentinel".

## Type

object

| Property                       | Type    | Description   |
|--------------------------------|---------|---|
| <code>affinity</code>          | object  | Affinity is a group of affinity scheduling rules.   |
| <code>certificate</code>       | object  | Certificate specifies the TLS certificate for Redis when EnableTLS is true.                     |
| <code>customConfig</code>      | object  | Config the config for sentinel  |
| <code>enableTLS</code>         | boolean | EnableTLS enable TLS for redis  |
| <code>exporter</code>          | object  | Exporter defines the specification for the sentinel exporter                                    |
| <code>expose</code>            | object  | Expose  |
| <code>externalTLSSecret</code> | string  | ExternalTLSSecret the external TLS secret to use, if not provided, the operator will create one |

| Property                      | Type                 | Description  |
|-------------------------------|----------------------|--|
| <code>image</code>            | <code>string</code>  | Image the redis sentinel image   |
| <code>imagePullPolicy</code>  | <code>string</code>  | ImagePullPolicy the image pull policy  |
| <code>imagePullSecrets</code> | <code>array</code>   | ImagePullSecrets the image pull secrets  |
| <code>monitorConfig</code>    | <code>object</code>  | <p>MonitorConfig configs for sentinel to monitor this replication, including:</p> <ul style="list-style-type: none"><li>• down-after-milliseconds</li><li>• failover-timeout</li><li>• parallel-syncs</li></ul>  |
| <code>nodeSelector</code>     | <code>object</code>  |  |
| <code>passwordSecret</code>   | <code>string</code>  | PasswordSecret   |
| <code>podAnnotations</code>   | <code>object</code>  |  |
| <code>quorum</code>           | <code>integer</code> | <p>Quorum the number of Sentinels that need to agree about the fact the master is not reachable, in order to really mark the master as failing, and eventually start a failover procedure if possible. If not specified, the default value is the majority of the Sentinels.</p> |

| Property                        | Type                 | Description   |
|---------------------------------|----------------------|---|
| <code>replicas</code>           | <code>integer</code> | Replicas the number of sentinel replicas  |
| <code>resources</code>          | <code>object</code>  | Resources the resources for sentinel  |
| <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>sentinelReference</code>  | <code>object</code>  | SentinelReference the sentinel reference  |
| <code>serviceAnnotations</code> | <code>object</code>  |   |
| <code>tolerations</code>        | <code>array</code>   |   |

## `.spec.sentinel.affinity`

### 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.sentinel.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 |

| Property   | Type          | Description  |
|--|---------------|--|
|  |               | <p>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 adding "weight" to the sum if the node matches the corresponding <code>matchExpressions</code>; the node(s) with the highest sum are the most preferred.</p> |
| <p><code>requiredDuringSchedulingIgnoredDuringExecution</code></p> | <p>object</p> | <p>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.</p>   |

## `.spec.sentinel.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.sentinel.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. |

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

## `.spec.sentinel.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.sentinel.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions`

### Description

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

### Type

`array`

# `.spec.sentinel.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.sentinel.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions`

## ns[].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.sentinel.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values[]**

### Type

string

## **.spec.sentinel.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields**

### Description

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

### Type

array

## **.spec.sentinel.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.sentinel.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.sentinel.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values[]`

### Type

string

## `.spec.sentinel.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> | array | Required. A list of node selector terms. The terms are ORed. |

## `.spec.sentinel.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms`

## Description

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

## Type

array

# `.spec.sentinel.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.sentinel.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions`

## Description

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

## Type

array

# `.spec.sentinel.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.  |
| 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.sentinel.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchE`

## xpressions[].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.sentinel.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values[]

### Type

string

## .spec.sentinel.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields

### Description

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

### Type

array

## .spec.sentinel.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.sentinel.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.sentinel.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values[]**

**Type**

string

**.spec.sentinel.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  |
|---|-------|--|
| preferredDuringSchedulingIgnoredDuringExecution | array | The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, 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 o |

| Property   | Type                      | Description  |
|--|---------------------------|--|
|  |                           | <p>scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution, affinity expressions, etc.) to compute a sum by iterating through the elements of this field adding "weight" to the sum if the node has a label which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.</p>  |
| <p><code>requiredDuringSchedulingIgnoredDuringExecution</code></p> | <p><code>array</code></p> | <p>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</p> |

| Property | Type | Description                                    |
|----------|------|--|
|          |      | intersected, i.e. all terms must be satisfied. |

## `.spec.sentinel.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.sentinel.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.sentinel.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 <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 <code>labelSelector</code> as <code>key in (value)</code> to select |

| Property                       | Type                | Description   |
|--------------------------------|---------------------|---|
|                                |                     | <p>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.</p>   |
| <code>mismatchLabelKeys</code> | <code>array</code>  | <p><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.</p> |
| <code>namespaceSelector</code> | <code>object</code> | <p>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.</p>  |
|                                |                     |   |

| 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.sentinel.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>  | <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.sentinel.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions`

### Description

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

### Type

`array`

## `.spec.sentinel.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.sentinel.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.sentinel.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector**

## `r.matchExpressions[].values[]`

### Type

string

## `.spec.sentinel.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.sentinel.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.

### Type

array

## `.spec.sentinel.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]`

### Type

string

## `.spec.sentinel.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.

### Type

array

## `.spec.sentinel.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]`

### Type

string

## `.spec.sentinel.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.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   |
|------------------|--------|---|
| 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.sentinel.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions`

## Description

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

## Type

array

## `.spec.sentinel.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.   |
| 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.sentinel.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.sentinel.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]**

#### Type

string

**.spec.sentinel.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.sentinel.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.sentinel.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces**  
**[]**

## Type

string

**.spec.sentinel.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.sentinel.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 <code>matchLabelKeys</code> and <code>labelSelector</code> . Also, <code>matchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. |
| 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 not 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   |

| Property          | Type   | Description  |
|-------------------|--------|--|
|                   |        | 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.  |
| 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.sentinel.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.sentinel.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

### Description

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

### Type

array

## `.spec.sentinel.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.sentinel.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.sentinel.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]**

#### Type

string

### **.spec.sentinel.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.sentinel.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.

#### Type

array

### `.spec.sentinel.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]`

#### Type

string

### `.spec.sentinel.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.

#### Type

array

### `.spec.sentinel.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]`

#### Type

string

## `.spec.sentinel.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  | <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.sentinel.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions`

### Description

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

### Type

array

## `.spec.sentinel.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.sentinel.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.sentinel.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]**

#### Type

string

### **.spec.sentinel.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.sentinel.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.sentinel.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]****Type**

string

**.spec.sentinel.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, 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, resource quota, etc.) and that does not violate any of the anti-affinity expressions, the scheduler picks the one that is most preferred. |

| Property  | Type         | Description  |
|---|--------------|--|
|   |              | <p>requiredDuringSchedulingIgnoredDuringExecution</p> <p>anti-affinity expression (e.g., compute a sum iterating through the elements of this field adding "weight" to the sum if the node has a label which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.</p>   |
| <p>requiredDuringSchedulingIgnoredDuringExecution</p> | <p>array</p> | <p>If the anti-affinity requirements specified in this field are not met at scheduling time, the pod will not be scheduled on 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</p> |

| Property | Type | Description                                    |
|----------|------|--|
|          |      | intersected, i.e. all terms must be satisfied. |

## `.spec.sentinel.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.sentinel.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 <code>podAffinityTerm</code> , in the range 1-100. |

## `.spec.sentinel.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 <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 <code>labelSelector</code> as <code>key in (value)</code> to select |

| Property                              | Type          | Description  |
|---------------------------------------|---------------|--|
|                                       |               | <p>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.</p>  |
| <p><code>mismatchLabelKeys</code></p> | <p>array</p>  | <p>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 <code>mismatchLabelKeys</code> and <code>labelSelector</code>. Also, <code>mismatchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set.</p> |
| <p><code>namespaceSelector</code></p> | <p>object</p> | <p>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.</p>   |
|                                       |               |  |

| 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.sentinel.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.sentinel.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions`

### Description

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

### Type

`array`

## `.spec.sentinel.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.sentinel.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.sentinel.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSe**

## lector.matchExpressions[].values[]

### Type

string

## .spec.sentinel.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.sentinel.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.

### Type

array

## `.spec.sentinel.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]`

### Type

string

## `.spec.sentinel.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.

### Type

array

## `.spec.sentinel.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]`

### Type

string

## `.spec.sentinel.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.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   |
|------------------|--------|---|
| 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.sentinel.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces.paceSelector.matchExpressions

## Description

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

## Type

array

## `.spec.sentinel.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.sentinel.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.sentinel.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces.selector.matchExpressions[].values[]**

#### Type

string

**.spec.sentinel.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces.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.sentinel.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.sentinel.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]**

### Type

string

## **.spec.sentinel.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.sentinel.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 <code>matchLabelKeys</code> and <code>labelSelector</code> . Also, <code>matchLabelKeys</code> cannot be set when <code>labelSelector</code> isn't set. |
| <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 not in</code>  |

| Property                       | Type                | Description  |
|--------------------------------|---------------------|--|
|                                |                     | <p><code>(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.</p> |
| <code>namespaceSelector</code> | <code>object</code> | <p>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.</p>  |
| <code>namespaces</code>        | <code>array</code>  | <p><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".</p>   |
|                                |                     |  |

| 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.sentinel.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> | <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.sentinel.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

### Description

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

### Type

array

## `.spec.sentinel.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. |
|          |        |  |

| 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.sentinel.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.sentinel.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]**

### **Type**

string

## **.spec.sentinel.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.sentinel.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.

### Type

array

## **.spec.sentinel.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]**

### Type

string

## **.spec.sentinel.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.

### Type

array

## `.spec.sentinel.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]`

### Type

string

## `.spec.sentinel.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  | 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 |

| Property | Type | Description  |
|----------|------|--|
|          |      | "key", the operator is "In", and the values array contains only "value". The requirements are ANDed. |

## `.spec.sentinel.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions`

### Description

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

### Type

array

## `.spec.sentinel.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. |

| Property              | Type                | Description  |
|-----------------------|---------------------|--|
| <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.sentinel.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.sentinel.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]`

### Type

`string`

## `.spec.sentinel.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.sentinel.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.sentinel.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]`

### Type

string

## `.spec.sentinel.certificate`

### Description

Certificate specifies the TLS certificate for Redis when EnableTLS is true.

## Type

object

| Property | Type   | Description  |
|----------|--------|--|
| dnsNames | array  | DNSNames is the list of DNS names to be included in the certificate  |
| duration | string | Requested 'duration' (i.e. lifetime) of the Certificate. Note that the issuer may choose to ignore the requested duration, just like any other requested attribute.<br><br>If unset, this defaults to 10 years. Minimum accepted duration is 1 hour. Value must be in units accepted by Go time.ParseDuration<br><a href="https://golang.org/pkg/time/#ParseDuration">https://golang.org/pkg/time/#ParseDuration</a> ↗ . |
| issuer   | object | Issuer is the issuer of the certificate  |

## .spec.sentinel.certificate.dnsNames

### Description

DNSNames is the list of DNS names to be included in the certificate

### Type

array

## .spec.sentinel.certificate.dnsNames[]

### Type

string

## .spec.sentinel.certificate.issuer

### Description

Issuer is the issuer of the certificate

### Type

object

| Property | Type   | Description         |
|----------|--------|---------------------|
| group    | string | Group of the Issuer |
| kind     | string | Kind of the Issuer  |
| name     | string | Name of the Issuer  |

## .spec.sentinel.customConfig

### Description

Config the config for sentinel

### Type

object

## .spec.sentinel.exporter

### Description

Exporter defines the specification for the sentinel exporter

### Type

object

| Property                     | Type                 | Description   |
|------------------------------|----------------------|---|
| <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 |

## .spec.sentinel.expose

### Description

Expose

### Type

`object`

| Property                            | Type                 | Description   |
|-------------------------------------|----------------------|---|
| <code>accessPort</code>             | <code>integer</code> | AccessPort defines the lb access nodeport   |
| <code>annotations</code>            | <code>object</code>  | The annotations of the service which attached to services   |
| <code>dataStorageNodePortMap</code> | <code>object</code>  | NodePortMap defines the map of the nodeport for redis sentinel only<br>Reversed for 3.14 backward compatibility |

| Property                                 | Type                 | Description  |
|--|----------------------|--|
| <code>dataStorageNodePortSequence</code> | <code>string</code>  | NodePortMap defines the map of the nodeport for redis nodes<br>NodePortSequence defines the sequence of the nodeport for redis cluster only                          |
| <code>enableNodePort</code>              | <code>boolean</code> | EnableNodePort defines if the nodeport is enabled  |
| <code>image</code>                       | <code>string</code>  | Image defines the image used to expose redis from annotations  |
| <code>imagePullPolicy</code>             | <code>string</code>  | ImagePullPolicy defines the image pull policy  |
| <code>ipFamilyPrefer</code>              | <code>string</code>  | 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. <code>service.spec.ipFamilies</code> ). |
| <code>type</code>                        | <code>string</code>  | ServiceType defines the type of the all related service  |

## `.spec.sentinel.expose.annotations`

### Description

The annotations of the service which attached to services

### Type

object

## **.spec.sentinel.expose.dataStorageNodePortMap**

### Description

NodePortMap defines the map of the nodeport for redis sentinel only Reversed for 3.14 backward compatibility

### Type

object

## **.spec.sentinel.imagePullSecrets**

### Description

ImagePullSecrets the image pull secrets

### Type

array

## **.spec.sentinel.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.<br>Instances of this type with an empty value here are almost |

| Property | Type | Description  |
|----------|------|--|
|          |      | certainly wrong. More info:<br><a href="https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names">https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names</a> ↗ |

## .spec.sentinel.monitorConfig

### Description

MonitorConfig configs for sentinel to monitor this replication, including: - down-after-milliseconds - failover-timeout - parallel-syncs

### Type

object

## .spec.sentinel.nodeSelector

### Type

object

## .spec.sentinel.podAnnotations

### Type

object

## .spec.sentinel.resources

### Description

Resources the resources for sentinel

### Type

object

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

## `.spec.sentinel.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.sentinel.resources.claims[]

### Description

ResourceClaim references one entry in PodSpec.ResourceClaims.

### Type

object

### Required

name

| Property             | Type                | Description   |
|----------------------|---------------------|---|
| <code>name</code>    | <code>string</code> | Name must match the name of one entry in <code>pod.spec.resourceClaims</code> of the Pod where this field is used. It makes that resource available inside a container. |
| <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.sentinel.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.sentinel.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.sentinel.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 | <p>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:</p> <ol style="list-style-type: none"> <li>1. The owning GID will be the FSGroup</li> <li>2. The setgid bit is set (new files created in the volume will be owned by FSGroup)</li> </ol> |

| Property                   | Type           | Description  |
|----------------------------|----------------|--|
|                            |                | <p>3. The permission bits are OR'd with rw-rw-<br/>---</p> <p>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.</p>  |
| <p>fsGroupChangePolicy</p> | <p>string</p>  | <p>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.</p> |
| <p>runAsGroup</p>          | <p>integer</p> | <p>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.</p>  |
|                            |                |  |

| Property                         | Type                 | Description  |
|----------------------------------|----------------------|--|
| <code>runAsNonRoot</code>        | <code>boolean</code> | <p>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.</p>                           |
| <code>runAsUser</code>           | <code>integer</code> | <p>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.</p>  |
| <code>seLinuxChangePolicy</code> | <code>string</code>  | <p><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".</p> <p>"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</p> |

| Property | Type | Description   |
|----------|------|---|
|          |      | <p>Pods sharing the same volume on the same node.</p> <p>"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.</p> <p>"MountOption" value is allowed only when SELinuxMount feature gate is enabled.</p> <p>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.</p> <p>This field affects only Pods that have SELinux label set, either in <code>PodSecurityContext</code> or in <code>SecurityContext</code> of all containers.</p> <p>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>.</p> |

| 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>  | <p>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 <code>supplementalGroupsPolicy</code> 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 <code>supplementalGroupsPolicy</code> field. Note that this field cannot be set when <code>spec.os.name</code> is windows.</p> |
| <code>supplementalGroupsPolicy</code> | <code>string</code> | <p>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 <code>SupplementalGroupsPolicy</code> feature gate to be enabled and the container runtime must implement support for this feature. Note that this field cannot be set when <code>spec.os.name</code> is windows.</p>  |
| <code>sysctls</code>                  | <code>array</code>  | <p>Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail</p>  |

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

## .spec.sentinel.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   |
|------------------|--------|---|
| 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:<br><br>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.sentinel.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.sentinel.securityContext.supplementalGroups[]`

### Type

integer

## `.spec.sentinel.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.sentinel.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.sentinel.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 ( <a href="https://github.com/kubernetes-sigs/windows-gmsa">https://github.com/kubernetes-sigs/windows-gmsa</a> ^) inlines the contents of the GMSA credential spec named by the GMSACredentialSpecName field. |
| gmsaCredentialSpecName | string | GMSACredentialSpecName is the name of the GMSA credential spec to use.  |

| Property                   | Type                 | Description  |
|----------------------------|----------------------|--|
| <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.sentinel.sentinelReference`

### Description

SentinelReference the sentinel reference

### Type

`object`

| Property          | Type                | Description            |
|-------------------|---------------------|------------------------|
| <code>auth</code> | <code>object</code> | Auth the sentinel auth |

| Property           | Type               | Description                      |
|--------------------|--------------------|----------------------------------|
| <code>nodes</code> | <code>array</code> | Addresses the sentinel addresses |

## `.spec.sentinel.sentinelReference.auth`

### Description

Auth the sentinel auth

### Type

`object`

| Property                    | Type                | Description                                  |
|-----------------------------|---------------------|--|
| <code>passwordSecret</code> | <code>string</code> | PasswordSecret the password secret for redis |
| <code>tlsSecret</code>      | <code>string</code> | TLSSecret the tls secret                     |
| <code>username</code>       | <code>string</code> | Username the username for redis              |

## `.spec.sentinel.sentinelReference.nodes`

### Description

Addresses the sentinel addresses

### Type

`array`

## `.spec.sentinel.sentinelReference.nodes[]`

## Type

object

| Property | Type    | Description                 |
|----------|---------|-----------------------------|
| flags    | string  | Flags                       |
| ip       | string  | IP the sentinel node ip     |
| port     | integer | Port the sentinel node port |

## .spec.sentinel.serviceAnnotations

### Type

object

## .spec.sentinel.tolerations

### Type

array

## .spec.sentinel.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.sentinelCustomConfig

### Description

SentinelCustomConfig defines custom Redis Sentinel configuration settings. Deprecated: Use sentinel.customConfig instead for improved organization.

### Type

object

## .spec.tolerations

### Description

Tolerations allow Redis pods to be scheduled on nodes with matching taints. See: <https://kubernetes.io/docs/concepts/scheduling-eviction/taint-and-toleration/>

### Type

array

## .spec.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. |

| Property                       | Type                 | Description   |
|--------------------------------|----------------------|---|
| <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.upgradeOption`

### Description

UpgradeOption defines the upgrade strategy and automation settings. Controls how and when Redis instances are upgraded.

### Type

object

| Property    | Type    | Description   |
|-------------|---------|---|
| autoUpgrade | boolean | AutoUpgrade controls whether the operator should automatically upgrade the instance when a new version becomes available. |
| crVersion   | string  | CRVersion specifies the target version for the Redis instance to upgrade to.  |

## .status

### Description

RedisStatus defines the observed state of a Redis instance.

### Type

object

| Property          | Type   | Description   |
|-------------------|--------|---|
| activeRedis       | object | ActiveRedis references the ActiveRedis instance configuration.  |
| clusterNodes      | array  | ClusterNodes contains detailed information about each Redis node in the cluster. Includes role, address, and health status. |
| detailedStatusRef | object | DetailedStatusRef references a resource with extended status information. Used for complex                                  |

| Property                        | Type                 | Description  |
|---------------------------------|----------------------|--|
|                                 |                      | status data that doesn't fit in the main status.   |
| <code>lastShardCount</code>     | <code>integer</code> | LastShardCount records the previous number of shards in the Redis Cluster. Used to detect and handle scaling operations.   |
| <code>lastVersion</code>        | <code>string</code>  | LastVersion records the previous version of the Redis instance. Used to detect and handle version upgrades.  |
| <code>matchLabels</code>        | <code>object</code>  | MatchLabels contains the selector labels used to identify Redis pods.  |
| <code>message</code>            | <code>string</code>  | Message provides additional details about the current status. Contains error information or in-progress operation details.   |
| <code>passwordSecretName</code> | <code>string</code>  | PasswordSecretName is the name of the Kubernetes Secret containing the Redis password. This is either user-specified or automatically generated.   |
| <code>phase</code>              | <code>string</code>  | Phase indicates the current operational state of the Redis instance. Values: Initializing - Resources are being created or reconciled Ready - All resources are ready and Redis is operational Error - An error occurred during resource initialization or operation |

| Property                      | Type                 | Description  |
|-------------------------------|----------------------|--|
|                               |                      | Rebalancing - Cluster slots are being redistributed<br>Paused - Reconciliation is paused   |
| <code>proxyMatchLabels</code> | <code>object</code>  | ProxyMatchLabels contains the selector labels used to identify Redis proxy pods.   |
| <code>proxyServiceName</code> | <code>string</code>  | ProxyServiceName is the name of the Kubernetes Service created for Redis Proxy.  |
| <code>restored</code>         | <code>boolean</code> | Restored indicates whether the instance has been successfully restored from a backup. Set to true once a restore operation completes successfully. |
| <code>serviceName</code>      | <code>string</code>  | ServiceName is the name of the main Kubernetes Service created for Redis access.   |
| <code>upgradeStatus</code>    | <code>object</code>  | UpgradeStatus contains detailed information about the ongoing or last upgrade operation.   |

## **.status.activeRedis**

### **Description**

ActiveRedis references the ActiveRedis instance configuration.

### **Type**

`object`

| Property               | Type                 | Description   |
|------------------------|----------------------|---|
| <code>name</code>      | <code>string</code>  | Name is the name of the ActiveRedis instance.   |
| <code>phase</code>     | <code>string</code>  | Phase indicates the current state of the ActiveRedis instance.                        |
| <code>serviceID</code> | <code>integer</code> | ServiceID is the unique identifier for this Redis instance in an active-active setup. |

## `.status.clusterNodes`

### Description

ClusterNodes contains detailed information about each Redis node in the cluster. Includes role, address, and health status.

### Type

`array`

## `.status.clusterNodes[]`

### Description

RedisNode represent a RedisCluster Node

### Type

`object`

### Required

`ip` `nodeName` `podName` `port` `role` `statefulSet`

| Property    | Type   | Description   |
|-------------|--------|---|
| id          | string | ID is the redis cluster node id, not runid  |
| ip          | string | IP is the ip of the node. if access announce is enabled, it will be the access ip       |
| masterRef   | string | MasterRef is the master node id of this node  |
| nodeName    | string | nodeName is the node name of the node where holds the pod                               |
| podName     | string | PodName current pod name  |
| port        | string | Port is the port of the node. if access announce is enabled, it will be the access port |
| role        | string | Role is the role of the node, master or slave   |
| slots       | array  | Slots is the slot range for the shard, eg: 0-1000,1002,1005-1100                        |
| statefulSet | string | StatefulSet is the statefulset name of this pod   |

## `.status.clusterNodes[].slots`

### Description

Slots is the slot range for the shard, eg: 0-1000,1002,1005-1100

### Type

array

## `.status.clusterNodes[].slots[]`

### Type

string

## `.status.detailedStatusRef`

### Description

DetailedStatusRef references a resource with extended status information. Used for complex status data that doesn't fit in the main status.

### Type

object

| Property                | Type                | Description   |
|-------------------------|---------------------|---|
| <code>apiVersion</code> | <code>string</code> | API version of the referent.  |
| <code>fieldPath</code>  | <code>string</code> | If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as <code>desiredState.manifest.containers[2]</code> . For example, if the object reference is to a container within a pod, this would take on a value like: <code>"spec.containers{name}"</code> (where "name" refers to the name of the container that triggered the event) or if no container name is |

| Property        | Type   | Description   |
|-----------------|--------|---|
|                 |        | specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.   |
| kind            | string | Kind of the referent. More info:<br><a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>   |
| name            | string | Name of the referent. More info:<br><a href="https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names">https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names</a>   |
| namespace       | string | Namespace of the referent. More info:<br><a href="https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/">https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/</a>  |
| resourceVersion | string | Specific resourceVersion to which this reference is made, if any. More info:<br><a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency</a> |
| uid             | string | UID of the referent. More info:<br><a href="https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids">https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids</a>  |

## .status.matchLabels

### Description

MatchLabels contains the selector labels used to identify Redis pods.

### Type

object

## .status.proxyMatchLabels

### Description

ProxyMatchLabels contains the selector labels used to identify Redis proxy pods.

### Type

object

## .status.upgradeStatus

### Description

UpgradeStatus contains detailed information about the ongoing or last upgrade operation.

### Type

object

| Property  | Type   | Description   |
|-----------|--------|---|
| crVersion | string | CRVersion indicates the target version for the upgrade operation.         |
| message   | string | Message provides additional information about the current upgrade status. |

# API Endpoints

The following API endpoints are available:

- `/apis/middleware.alauda.io/v1/namespaces/{namespace}/redis`
  - `DELETE` : delete collection of Redis
  - `GET` : list objects of kind Redis
  - `POST` : create a new Redis
- `/apis/middleware.alauda.io/v1/namespaces/{namespace}/redis/{name}`
  - `DELETE` : delete the specified Redis
  - `GET` : read the specified Redis
  - `PATCH` : partially update the specified Redis
  - `PUT` : replace the specified Redis
- `/apis/middleware.alauda.io/v1/namespaces/{namespace}/redis/{name}/status`
  - `GET` : read status of the specified Redis
  - `PATCH` : partially update status of the specified Redis
  - `PUT` : replace status of the specified Redis

## `/apis/middleware.alauda.io/v1/namespaces/{namespace}/redis`

### HTTP method

`DELETE`

### Description

delete collection of Redis

### HTTP responses

| HTTP code          | Response body              |
|--------------------|----------------------------|
| 200 - OK           | <code>Status</code> schema |
| 401 - Unauthorized | Empty                      |

## HTTP method

GET

## Description

list objects of kind Redis

## HTTP responses

| HTTP code          | Response body                 |
|--------------------|-------------------------------|
| 200 - OK           | <code>RedisList</code> schema |
| 401 - Unauthorized | Empty                         |

## HTTP method

POST

## Description

create a new Redis

## 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 |

| 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      | Redis schema | application/json formatted |

### HTTP responses

| HTTP code          | Response body |
|--------------------|---------------|
| 200 - OK           | Redis schema  |
| 201 - Created      | Redis schema  |
| 202 - Accepted     | Redis schema  |
| 401 - Unauthorized | Empty         |

**/apis/middleware.alauda.io/v1/namespaces/{namespace}/redis/{name}**

### HTTP method

DELETE

## Description

delete the specified Redis

## 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 Redis

## HTTP responses

| HTTP code          | Response body             |
|--------------------|---------------------------|
| 200 - OK           | <code>Redis</code> schema |
| 401 - Unauthorized | Empty                     |

## HTTP method

`PATCH`

## Description

partially update the specified Redis

## 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>Redis</code> schema |
| 401 - Unauthorized | Empty                     |

## HTTP method

`PUT`

## Description

replace the specified Redis

## 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>Redis</code> schema | <code>application/json</code> formatted |

## HTTP responses

| HTTP code          | Response body             |
|--------------------|---------------------------|
| 200 - OK           | <code>Redis</code> schema |
| 201 - Created      | <code>Redis</code> schema |
| 401 - Unauthorized | Empty                     |

## /apis/middleware.alauda.io/v1/namespaces/{namespace}/redis/{name}/status

### HTTP method

`GET`

### Description

read status of the specified Redis

### HTTP responses

| HTTP code          | Response body             |
|--------------------|---------------------------|
| 200 - OK           | <code>Redis</code> schema |
| 401 - Unauthorized | Empty                     |

### HTTP method

`PATCH`

### Description

partially update status of the specified Redis

### 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 |

| Parameter                    | Type                | Description  |
|------------------------------|---------------------|--|
|                              |                     | processing of the request. Valid values are: - All: all dry run stages will be processed   |
| <code>fieldValidation</code> | <code>string</code> | <p><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.</p> |

## HTTP responses

| HTTP code          | Response body             |
|--------------------|---------------------------|
| 200 - OK           | <code>Redis</code> schema |
| 401 - Unauthorized | Empty                     |

## HTTP method

`PUT`

## Description

replace status of the specified Redis

## 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>Redis</code> schema | <code>application/json</code> formatted |

## HTTP responses

| HTTP code     | Response body             |
|---------------|---------------------------|
| 200 - OK      | <code>Redis</code> schema |
| 201 - Created | <code>Redis</code> schema |

| HTTP code          | Response body |
|--------------------|---------------|
| 401 - Unauthorized | Empty         |

# Redis 用户

## Description

RedisUser is the Schema for the redisusers API. Represents a Redis user account with associated ACL permissions.

## 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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a> |
| 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:  |

| Property | Type       | Description   |
|----------|------------|---|
|          |            | <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a> |
| metadata | ObjectMeta | ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.  |
| spec     | object     | RedisUserSpec defines the desired state of RedisUser  |
| status   | object     | RedisUserStatus defines the observed state of RedisUser   |

## .spec

### Description

RedisUserSpec defines the desired state of RedisUser

### Type

object

### Required

redisName

username

| Property    | Type   | Description  |
|-------------|--------|--|
| accountType | string | AccountType defines the user account category.<br>Options: system (privileged system accounts), custom (general user accounts), default (default user configuration) |

| Property                     | Type                | Description  |
|------------------------------|---------------------|--|
| <code>aclRules</code>        | <code>string</code> | AclRules defines the Redis ACL rules as a string. Format follows Redis ACL syntax for permissions and command restrictions. See: <a href="https://redis.io/docs/management/security/acl/">https://redis.io/docs/management/security/acl/</a> |
| <code>arch</code>            | <code>string</code> | Arch specifies the Redis architecture type for this user. Must match the architecture of the referenced Redis instance.  |
| <code>passwordSecrets</code> | <code>array</code>  | PasswordSecrets is a list of Kubernetes Secret names containing user passwords. Each secret must have a 'password' key containing the Redis password. Multiple secrets allow for credential rotation.  |
| <code>redisName</code>       | <code>string</code> | RedisName specifies the target Redis instance name. Must reference an existing Redis resource in the same namespace.   |
| <code>username</code>        | <code>string</code> | Username specifies the Redis user account name. Required for user identification during authentication.  |

## .spec.passwordSecrets

### Description

PasswordSecrets is a list of Kubernetes Secret names containing user passwords. Each secret must have a 'password' key containing the Redis password. Multiple secrets allow for credential rotation.

## Type

array

## .spec.passwordSecrets[]

## Type

string

## .status

## Description

RedisUserStatus defines the observed state of RedisUser

## Type

object

| Property          | Type   | Description   |
|-------------------|--------|---|
| Phase             | string | Phase indicates the current state of the RedisUser resource. Values: Fail, Success, Pending                             |
| aclRules          | string | AclRules reflects the actual ACL rules applied to Redis.  |
| lastUpdateSuccess | string | LastUpdatedSuccess records the timestamp of the last successful configuration update. Format: RFC3339 timestamp string. |
|                   |        |   |

| Property | Type   | Description  |
|----------|--------|--|
| message  | string | Message provides additional details about the current status. Contains error information or operation details. |

## API Endpoints

The following API endpoints are available:

- `/apis/redis.middleware.alauda.io/v1/namespaces/{namespace}/redisusers`
  - **DELETE** : delete collection of RedisUser
  - **GET** : list objects of kind RedisUser
  - **POST** : create a new RedisUser
- `/apis/redis.middleware.alauda.io/v1/namespaces/{namespace}/redisusers/{name}`
  - **DELETE** : delete the specified RedisUser
  - **GET** : read the specified RedisUser
  - **PATCH** : partially update the specified RedisUser
  - **PUT** : replace the specified RedisUser
- `/apis/redis.middleware.alauda.io/v1/namespaces/{namespace}/redisusers/{name}/status`
  - **GET** : read status of the specified RedisUser
  - **PATCH** : partially update status of the specified RedisUser
  - **PUT** : replace status of the specified RedisUser

# /apis/redis.middleware.alauda.io/v1/namespaces/{namespace}/redisusers

## HTTP method

DELETE

## Description

delete collection of RedisUser

## HTTP responses

| HTTP code          | Response body                 |
|--------------------|-------------------------------|
| 200 - OK           | <a href="#">Status</a> schema |
| 401 - Unauthorized | Empty                         |

## HTTP method

GET

## Description

list objects of kind RedisUser

## HTTP responses

| HTTP code          | Response body                        |
|--------------------|--------------------------------------|
| 200 - OK           | <a href="#">RedisUserList</a> schema |
| 401 - Unauthorized | Empty                                |

## HTTP method

POST

## Description

create a new RedisUser

## 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>RedisUser</code> schema | <code>application/json</code> formatted |

## HTTP responses

| HTTP code     | Response body                 |
|---------------|-------------------------------|
| 200 - OK      | <code>RedisUser</code> schema |
| 201 - Created | <code>RedisUser</code> schema |

| HTTP code          | Response body                 |
|--------------------|-------------------------------|
| 202 - Accepted     | <code>RedisUser</code> schema |
| 401 - Unauthorized | Empty                         |

## /apis/redis.middleware.alauda.io/v1/namespaces/{namespace}/redisusers/{name}

### HTTP method

DELETE

### Description

delete the specified RedisUser

### 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 RedisUser

## HTTP responses

| HTTP code          | Response body                 |
|--------------------|-------------------------------|
| 200 - OK           | <code>RedisUser</code> schema |
| 401 - Unauthorized | Empty                         |

## HTTP method

`PATCH`

## Description

partially update the specified RedisUser

## 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 |

| Parameter | Type | Description  |
|-----------|------|--|
|           |      | will contain all unknown and duplicate fields encountered. |

## HTTP responses

| HTTP code          | Response body                 |
|--------------------|-------------------------------|
| 200 - OK           | <code>RedisUser</code> schema |
| 401 - Unauthorized | Empty                         |

## HTTP method

PUT

## Description

replace the specified RedisUser

## 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 |

| Parameter | Type | Description   |
|-----------|------|---|
|           |      | 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      | RedisUser schema | application/json formatted |

### HTTP responses

| HTTP code          | Response body    |
|--------------------|------------------|
| 200 - OK           | RedisUser schema |
| 201 - Created      | RedisUser schema |
| 401 - Unauthorized | Empty            |

## /apis/redis.middleware.alauda.io/v1/namespaces/{namespace}/redisusers/{name}/status

### HTTP method

GET

### Description

read status of the specified RedisUser

### HTTP responses

| HTTP code | Response body    |
|-----------|------------------|
| 200 - OK  | RedisUser schema |

| HTTP code          | Response body |
|--------------------|---------------|
| 401 - Unauthorized | Empty         |

## HTTP method

PATCH

## Description

partially update status of the specified RedisUser

## 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>RedisUser</code> schema |
| 401 - Unauthorized | Empty                         |

## HTTP method

PUT

## Description

replace status of the specified RedisUser

## 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                |
|-----------|------------------|----------------------------|
| body      | RedisUser schema | application/json formatted |

## HTTP responses

| HTTP code          | Response body    |
|--------------------|------------------|
| 200 - OK           | RedisUser schema |
| 201 - Created      | RedisUser schema |
| 401 - Unauthorized | Empty            |

# ActiveRedisConnection

## Description

ActiveRedisConnection is the Schema for the activeredisconnections API

## Type

object

## Specification

| Property                | Type                | Description  |
|-------------------------|---------------------|--|
| <code>apiVersion</code> | <code>string</code> | <p>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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a></p> |

| 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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a> |
| metadata | ObjectMeta | ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.   |
| spec     | object     | ActiveRedisConnectionSpec defines the desired state of ActiveRedisConnection   |
| status   | object     | ActiveRedisConnectionStatus defines the observed state of ActiveRedisConnection  |

## .spec

### Description

ActiveRedisConnectionSpec defines the desired state of ActiveRedisConnection

### Type

object

### Required

addresses

instance

secretName

| Property                     | Type                 | Description   |
|------------------------------|----------------------|---|
| <code>activeRedisName</code> | <code>string</code>  | DeActiveRedisName is the name of the readonly active redis.   |
| <code>addresses</code>       | <code>array</code>   | Addresses is a list of redis addresses for the active redis connection. It can be a single address or multiple addresses.   |
| <code>enableTLS</code>       | <code>boolean</code> | EnableTLS enable tls for redis connection.  |
| <code>instance</code>        | <code>string</code>  | Instance is the instance name of local redis.   |
| <code>pause</code>           | <code>boolean</code> | Pause is used to pause the active redis connection. When set to true, the active redis connection will not be established. This is useful for maintenance or temporary disabling of the connection. |
| <code>secretName</code>      | <code>string</code>  | SecretName the secret name for redis authentication. for default user, the password field is required.  |

## `.spec.addresses`

### Description

Addresses is a list of redis addresses for the active redis connection. It can be a single address or multiple addresses.

### Type

`array`

## `.spec.addresses[]`

### Type

`string`

## `.status`

### Description

ActiveRedisConnectionStatus defines the observed state of ActiveRedisConnection

### Type

`object`

| Property                     | Type                | Description  |
|------------------------------|---------------------|--|
| <code>activeRedisName</code> | <code>string</code> | ActiveRedisName  |
| <code>instance</code>        | <code>string</code> | Instance is the instance name of local redis.  |
| <code>message</code>         | <code>string</code> | Message  |
| <code>network</code>         | <code>object</code> | Network is the network status for redis connection.  |
| <code>shards</code>          | <code>array</code>  | Role string <code>json:"role,omitempty"</code> Shards is the shards status for peer redis. |

| Property                  | Type                | Description  |
|---------------------------|---------------------|--|
| <code>status</code>       | <code>string</code> | Status is the status of the active redis connection. |
| <code>upstreamPeer</code> | <code>object</code> | UpstreamPeer is the upstream peer redis.             |

## `.status.network`

### Description

Network is the network status for redis connection.

### Type

`object`

| Property                   | Type                 | Description                                       |
|----------------------------|----------------------|---|
| <code>delay</code>         | <code>integer</code> | Delay is the network delay for ping in ms.        |
| <code>lastCheckTime</code> | <code>integer</code> | LastCheckTime is the last check time for network. |

## `.status.shards`

### Description

Role string `json:"role,omitempty"` Shards is the shards status for peer redis.

### Type

`array`

## `.status.shards[]`

## Type

object

## Required

index

status

syncStatus

| Property   | Type    | Description  |
|------------|---------|--|
| index      | integer | Index is the index of the shard.                   |
| offset     | string  | Offset is the current offset sync from peer redis. |
| opId       | string  | OpId is the current opId sync from peer redis.     |
| status     | string  | Status is the status of the shard.                 |
| syncStatus | string  | SyncStatus is the sync status of the shard.        |

## .status.upstreamPeer

### Description

UpstreamPeer is the upstream peer redis.

### Type

object

| Property                      | Type                 | Description   |
|-------------------------------|----------------------|---|
| <code>service_id</code>       | <code>integer</code> | ServiceID is the service ID of the upstream peer redis.             |
| <code>service_metadata</code> | <code>object</code>  | ServiceMetadata is the metadata of the upstream peer redis service. |

## `.status.upstreamPeer.service_metadata`

### Description

ServiceMetadata is the metadata of the upstream peer redis service.

### Type

`object`

## API Endpoints

The following API endpoints are available:

- `/apis/redis.middleware.alauda.io/v1alpha1/namespaces/{namespace}/activeredisconnections`
  - `DELETE` : delete collection of ActiveRedisConnection
  - `GET` : list objects of kind ActiveRedisConnection
  - `POST` : create a new ActiveRedisConnection
- `/apis/redis.middleware.alauda.io/v1alpha1/namespaces/{namespace}/activeredisconnections/{name}`
  - `DELETE` : delete the specified ActiveRedisConnection
  - `GET` : read the specified ActiveRedisConnection

- **PATCH** : partially update the specified ActiveRedisConnection
- **PUT** : replace the specified ActiveRedisConnection
- `/apis/redis.middleware.alauda.io/v1alpha1/namespaces/{namespace}/activeredisconnections/{name}/status`
  - **GET** : read status of the specified ActiveRedisConnection
  - **PATCH** : partially update status of the specified ActiveRedisConnection
  - **PUT** : replace status of the specified ActiveRedisConnection

## `/apis/redis.middleware.alauda.io/v1alpha1/namespaces/{namespace}/activeredisconnections`

### HTTP method

**DELETE**

### Description

delete collection of ActiveRedisConnection

### HTTP responses

| HTTP code          | Response body              |
|--------------------|----------------------------|
| 200 - OK           | <code>Status</code> schema |
| 401 - Unauthorized | Empty                      |

### HTTP method

**GET**

### Description

list objects of kind ActiveRedisConnection

### HTTP responses

| HTTP code | Response body                                 |
|-----------|---|
| 200 - OK  | <code>ActiveRedisConnectionList</code> schema |

| HTTP code          | Response body |
|--------------------|---------------|
| 401 - Unauthorized | Empty         |

## HTTP method

POST

## Description

create a new ActiveRedisConnection

## 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                |
|-----------|------------------------------|----------------------------|
| body      | ActiveRedisConnection schema | application/json formatted |

## HTTP responses

| HTTP code          | Response body                |
|--------------------|------------------------------|
| 200 - OK           | ActiveRedisConnection schema |
| 201 - Created      | ActiveRedisConnection schema |
| 202 - Accepted     | ActiveRedisConnection schema |
| 401 - Unauthorized | Empty                        |

## /apis/redis.middleware.alauda.io/v1alpha1/namespaces/{namespace}/activeredisconnections/{name}

### HTTP method

DELETE

### Description

delete the specified ActiveRedisConnection

### 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 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 ActiveRedisConnection

## HTTP responses

| HTTP code          | Response body                             |
|--------------------|---|
| 200 - OK           | <code>ActiveRedisConnection</code> schema |
| 401 - Unauthorized | Empty                                     |

## HTTP method

PATCH

## Description

partially update the specified ActiveRedisConnection

## 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:  |

| Parameter | Type | Description  |
|-----------|------|--|
|           |      | <p>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.</p> |

## HTTP responses

| HTTP code          | Response body                             |
|--------------------|---|
| 200 - OK           | <code>ActiveRedisConnection</code> schema |
| 401 - Unauthorized | Empty                                     |

## HTTP method

PUT

## Description

replace the specified ActiveRedisConnection

## Query parameters

| Parameter           | Type                | Description   |
|---------------------|---------------------|---|
| <code>dryRun</code> | <code>string</code> | <p>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</p> |

| Parameter                    | Type                | Description   |
|------------------------------|---------------------|---|
| <code>fieldValidation</code> | <code>string</code> | <p><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:</p> <ul style="list-style-type: none"> <li>- 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.</li> <li>- 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+.</li> <li>- 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.</li> </ul> |

## Body parameters

| Parameter         | Type                                      | Description                             |
|-------------------|---|---|
| <code>body</code> | <code>ActiveRedisConnection</code> schema | <code>application/json</code> formatted |

## HTTP responses

| HTTP code          | Response body                             |
|--------------------|---|
| 200 - OK           | <code>ActiveRedisConnection</code> schema |
| 201 - Created      | <code>ActiveRedisConnection</code> schema |
| 401 - Unauthorized | Empty                                     |

**`/apis/redis.middleware.alauda.io/v1alpha1/namespaces/{namespace}/activeredisconnections/{name}/status`**

## HTTP method

GET

## Description

read status of the specified ActiveRedisConnection

## HTTP responses

| HTTP code          | Response body                             |
|--------------------|---|
| 200 - OK           | <code>ActiveRedisConnection</code> schema |
| 401 - Unauthorized | Empty                                     |

## HTTP method

PATCH

## Description

partially update status of the specified ActiveRedisConnection

## 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. |

| Parameter | Type | Description  |
|-----------|------|--|
|           |      | 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>ActiveRedisConnection</code> schema |
| 401 - Unauthorized | Empty                                     |

## HTTP method

PUT

## Description

replace status of the specified ActiveRedisConnection

## 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, |

| Parameter | Type | Description   |
|-----------|------|---|
|           |      | 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      | ActiveRedisConnection schema | application/json formatted |

## HTTP responses

| HTTP code          | Response body                |
|--------------------|------------------------------|
| 200 - OK           | ActiveRedisConnection schema |
| 201 - Created      | ActiveRedisConnection schema |
| 401 - Unauthorized | Empty                        |