Storage APIs

PersistentVolumeClaim [v1]

PersistentVolumeClaim [v1]

PersistentVolume [v1]

PersistentVolume [v1]

StorageClass [storage.k8s.io/v1]

StorageClass [storage.k8s.io/v1]

Ceph Storage APIs

CephCluster [cephclusters.cep | CephFilesystem [cephfilesyste | CephBlockI

CephObjectStore [cephobjectst | CephObjectStoreUser [cephobjectstoreusers.

PersistentVolumeClaim [v1]

描述

PersistentVolumeClaim is a user's request for and claim to a persistent volume

类型

`object`

规格

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

属性	类型	描述
`kind`	`string`	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`metadata`	`ObjectMeta`	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
`spec`	`object`	PersistentVolumeClaimSpec describes the common attributes of storage devices and allows a Source for provider-specific attributes
`status`	`object`	PersistentVolumeClaimStatus is the current status of a persistent volume claim.

.spec

描述

PersistentVolumeClaimSpec describes the common attributes of storage devices and allows a Source for provider-specific attributes

类型

`object`

属性	类型	描述
`accessModes`	`array`	accessModes contains the desired access mod volume should have. More info: https://kubernetes.io/docs/concepts/storage/per volumes#access-modes-1 ↗
`dataSource`	`object`	TypedLocalObjectReference contains enough information to let you locate the typed reference object inside the same namespace.
`dataSourceRef`	`object`	TypedObjectReference contains enough inform let you locate the typed referenced object
`resources`	`object`	VolumeResourceRequirements describes the s resource requirements for a volume.
`selector`	`object`	A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label s matches all objects. A null label selector matche objects.
`storageClassName`	`string`	storageClassName is the name of the StorageC required by the claim. More info: https://kubernetes.io/docs/concepts/storage/per volumes#class-1 ↗

属性	类型	描述
`volumeAttributesClassName`	`string`	volumeAttributesClassName may be used to se VolumeAttributesClass used by this claim. If spe the CSI driver will create or update the volume attributes defined in the corresponding VolumeAttributesClass. This has a different pur than storageClassName, it can be changed afte claim is created. An empty string value means t VolumeAttributesClass will be applied to the cla it's not allowed to reset this field to empty string is set. If unspecified and the PersistentVolumeC unbound, the default VolumeAttributesClass wil by the persistentvolume controller if it exists. If t resource referred to by volumeAttributesClass d not exist, this PersistentVolumeClaim will be se Pending state, as reflected by the modifyVolum field, until such as a resource exists. More info: https://kubernetes.io/docs/concepts/storage/vol attributes-classes/↗ (Beta) Using this field requ the VolumeAttributesClass feature gate to be en (off by default).
`volumeMode`	`string`	volumeMode defines what type of volume is req by the claim. Value of Filesystem is implied whe included in claim spec. Possible enum values: • `"Block"` means the volume will not be forr with a filesystem and will remain a raw block device. • `"Filesystem"` means the volume will be d formatted with a filesystem.

属性	类型	描述
`volumeName`	`string`	volumeName is the binding reference to the PersistentVolume backing this claim.

.spec.accessModes

描述

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

类型

`array`

.spec.accessModes[]

类型

`string`

.spec.dataSource

描述

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

类型

`object`

必填

`kind` `name`

属性	类型	描述
`apiGroup`	`string`	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
`kind`	`string`	Kind is the type of resource being referenced
`name`	`string`	Name is the name of resource being referenced

.spec.dataSourceRef

描述

TypedObjectReference contains enough information to let you locate the typed referenced object

类型

`object`

必填

`kind` `name`

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

属性	类型	描述
`kind`	`string`	Kind is the type of resource being referenced
`name`	`string`	Name is the name of resource being referenced
`namespace`	`string`	Namespace is the namespace of resource being referenced Note that when a namespace is specified, a gateway.networking.k8s.io/ReferenceGrant object is required in the referent namespace to allow that namespace's owner to accept the reference. See the ReferenceGrant documentation for details. (Alpha) This field requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

.spec.resources

描述

VolumeResourceRequirements describes the storage resource requirements for a volume.

类型

`object`

属性	类型	描述
`limits`	`object`	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

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

.spec.resources.limits

描述

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

类型

`object`

.spec.resources.requests

描述

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/

类型

`object`

.spec.selector

描述

A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

类型

`object`

属性	类型	描述
`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.selector.matchExpressions

描述

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

类型

`array`

.spec.selector.matchExpressions[]

描述

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

类型

`object`

必填

`key` `operator`

属性	类型	描述
`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.selector.matchExpressions[].values

描述

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.

类型

`array`

.spec.selector.matchExpressions[].values[]

类型

`string`

.spec.selector.matchLabels

描述

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.

类型

`object`

.status

描述

PersistentVolumeClaimStatus is the current status of a persistent volume claim.

类型

`object`

属性	类型	描述
`accessModes`	`array`	accessModes contains the actual acce volume backing the PVC has. More inf https://kubernetes.io/docs/concepts/sto volumes#access-modes-1 ↗
`allocatedResourceStatuses`	`object`	allocatedResourceStatuses stores stat being resized for the given PVC. Key n standard Kubernetes label syntax. Vali either: * Un-prefixed keys: - storage - t the volume. * Custom resources must u implementation-defined prefixed name "example.com/my-custom-resource" Ap values - keys that are unprefixed or ha kubernetes.io prefix are considered res hence may not be used.

属性	类型	描述
		ClaimResourceStatus can be in any of states: - ControllerResizeInProgress: S resize controller starts resizing the volu plane. - ControllerResizeFailed: State s has failed in resize controller with a terr NodeResizePending: State set when re has finished resizing the volume but fur volume is needed on the node. - NodeResizeInProgress: State set wher resizing the volume. - NodeResizeFaile when resizing has failed in kubelet with error. Transient errors don't set NodeRe example: if expanding a PVC for more field can be one of the following states: pvc.status.allocatedResourceStatus['st "ControllerResizeInProgress" - pvc.status.allocatedResourceStatus['st "ControllerResizeFailed" - pvc.status.allocatedResourceStatus['st "NodeResizePending" - pvc.status.allocatedResourceStatus['st "NodeResizeInProgress" - pvc.status.allocatedResourceStatus['st "NodeResizeFailed" When this field is r means that no resize operation is in pro given PVC. A controller that receives PVC update w unknown resourceName or ClaimReso should ignore the update for the purpos designed. For example - a controller tha responsible for resizing capacity of the ignore PVC updates that change other associated with PVC.

属性	类型	描述
		This is an alpha field and requires enab RecoverVolumeExpansionFailure featu

属性	类型	描述
`allocatedResources`	`object`	allocatedResources tracks the resource a PVC including its capacity. Key name standard Kubernetes label syntax. Valid either: * Un-prefixed keys: - storage - th the volume. * Custom resources must u implementation-defined prefixed names "example.com/my-custom-resource" Ap values - keys that are unprefixed or hav kubernetes.io prefix are considered res hence may not be used. Capacity reported here may be larger th capacity when a volume expansion ope requested. For storage quota, the large allocatedResources and PVC.spec.res If allocatedResources is not set, PVC.s alone is used for quota calculation. If a expansion capacity request is lowered, allocatedResources is only lowered if th expansion operations in progress and i volume capacity is equal or lower than capacity. A controller that receives PVC update w unknown resourceName should ignore the purpose it was designed. For exam controller that only is responsible for re of the volume, should ignore PVC upda change other valid resources associate This is an alpha field and requires enab RecoverVolumeExpansionFailure featu

属性	类型	描述
`capacity`	`object`	capacity represents the actual resource underlying volume.
`conditions`	`array`	conditions is the current Condition of pe volume claim. If underlying persistent v resized then the Condition will be set to
`currentVolumeAttributesClassName`	`string`	currentVolumeAttributesClassName is t name of the VolumeAttributesClass the When unset, there is no VolumeAttribut to this PersistentVolumeClaim This is a requires enabling VolumeAttributesClas by default).
`modifyVolumeStatus`	`object`	ModifyVolumeStatus represents the sta ControllerModifyVolume operation
`phase`	`string`	phase represents the current phase of PersistentVolumeClaim. Possible enum values: - `"Bound"` used for PersistentVolum are bound - `"Lost"` used for PersistentVolume lost their underlying PersistentVolun was bound to a PersistentVolume a does not exist any longer and all da lost.

属性	类型	描述
		• `"Pending"` used for PersistentVol are not yet bound

.status.accessModes

描述

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

类型

`array`

.status.accessModes[]

类型

`string`

.status.allocatedResourceStatuses

描述

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

capacity - this field can be one of the following states: - pvc.status.allocatedResourceStatus['storage'] = "ControllerResizeInProgress" - pvc.status.allocatedResourceStatus['storage'] = "ControllerResizeFailed" - pvc.status.allocatedResourceStatus['storage'] = "NodeResizePending" - pvc.status.allocatedResourceStatus['storage'] = "NodeResizeInProgress" - pvc.status.allocatedResourceStatus['storage'] = "NodeResizeFailed" When this field is not set, it means that no resize operation is in progress for the given PVC. A controller that receives PVC update with previously unknown resourceName or ClaimResourceStatus should ignore the update for the purpose it was designed. For example - a controller that only is responsible for resizing capacity of the volume, should ignore PVC updates that change other valid resources associated with PVC. This is an alpha field and requires enabling RecoverVolumeExpansionFailure feature.

类型

`object`

.status.allocatedResources

描述

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

类型

`object`

.status.capacity

描述

capacity represents the actual resources of the underlying volume.

类型

`object`

.status.conditions

描述

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

类型

`array`

.status.conditions[]

描述

PersistentVolumeClaimCondition contains details about state of pvc

类型

`object`

必填

`type` `status`

属性	类型	描述
`lastProbeTime`	`string`	Time is a wrapper around time.Time which supports cor JSON. Wrappers are provided for many of the factory m offers.

属性	类型	描述
`lastTransitionTime`	`string`	Time is a wrapper around time.Time which supports cor JSON. Wrappers are provided for many of the factory m offers.
`message`	`string`	message is the human-readable message indicating de
`reason`	`string`	reason is a unique, this should be a short, machine und the reason for condition's last transition. If it reports "Re underlying persistent volume is being resized.
`status`	`string`	Status is the status of the condition. Can be True, False https://kubernetes.io/docs/reference/kubernetes-api/con resources/persistent-volume-claim-v1/#:~:text=state%20of%20pvc-,conditions.status,-(strin
`type`	`string`	Type is the type of the condition. More info: https://kubernetes.io/docs/reference/kubernetes-api/con resources/persistent-volume-claim-v1/#:~:text=set%20to%20%27ResizeStarted%27.-,Pers contains%20details%20about ↗

.status.modifyVolumeStatus

描述

ModifyVolumeStatus represents the status object of ControllerModifyVolume operation

类型

`object`

必填

`status`

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

属性	类型	描述
		• `"Infeasible"` Infeasible indicates that the request has been rejected as invalid by the CSI driver. To resolve the error, a valid VolumeAttributesClass needs to be specified • `"Pending"` Pending indicates that the PersistentVolumeClaim cannot be modified due to unmet requirements, such as the specified VolumeAttributesClass not existing
`targetVolumeAttributesClassName`	`string`	targetVolumeAttributesClassName is the name of the VolumeAttributesClass the PVC currently being reconciled

API 端点

可用的 API 端点如下：

- `/kubernetes/{cluster}/api/v1/namespaces/{namespace}/persistentvolumeclaims`
  - `DELETE`: delete collection of PersistentVolumeClaim
  - `GET`: list objects of kind PersistentVolumeClaim
  - `POST`: create a new PersistentVolumeClaim
- `/kubernetes/{cluster}/api/v1/namespaces/{namespace}/persistentvolumeclaims/{name}`

- `DELETE`: delete the specified PersistentVolumeClaim
- `GET`: read the specified PersistentVolumeClaim
- `PATCH`: partially update the specified PersistentVolumeClaim
- `PUT`: replace the specified PersistentVolumeClaim

- `/kubernetes/{cluster}/api/v1/namespaces/{namespace}/persistentvolumeclaims/{name}/status`
  - `GET`: read status of the specified PersistentVolumeClaim
  - `PATCH`: partially update status of the specified PersistentVolumeClaim
  - `PUT`: replace status of the specified PersistentVolumeClaim

/kubernetes/{cluster}/api/v1/namespaces/{namespace}/persistentvolumeclaims

HTTP 方法

`DELETE`

描述

delete collection of PersistentVolumeClaim

HTTP 响应

HTTP 状态码	响应体
200 - OK	`Status` schema
401 - Unauthorized	Empty

HTTP 方法

`GET`

描述

list objects of kind PersistentVolumeClaim

HTTP 响应

HTTP 状态码	响应体
200 - OK	`PersistentVolumeClaimList` schema
401 - Unauthorized	Empty

HTTP 方法

`POST`

描述

create a new PersistentVolumeClaim

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	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.

请求体参数

参数	类型	描述
`body`	`PersistentVolumeClaim` schema	`application/json` formatted

HTTP 响应

HTTP 状态码	响应体
200 - OK	`PersistentVolumeClaim` schema
201 - Created	`PersistentVolumeClaim` schema
202 - Accepted	`PersistentVolumeClaim` schema
401 - Unauthorized	Empty

/kubernetes/{cluster}/api/v1/namespaces/{namespace}/persistentvolumeclaims/{name}

HTTP 方法

`DELETE`

描述

delete the specified PersistentVolumeClaim

查询参数

参数	类型	描述
`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 响应

HTTP 状态码	响应体
200 - OK	`Status` schema
202 - Accepted	`Status` schema
401 - Unauthorized	Empty

HTTP 方法

`GET`

描述

read the specified PersistentVolumeClaim

HTTP 响应

HTTP 状态码	响应体
200 - OK	`PersistentVolumeClaim` schema
401 - Unauthorized	Empty

HTTP 方法

`PATCH`

描述

partially update the specified PersistentVolumeClaim

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore:

参数	类型	描述
		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.

HTTP 响应

HTTP 状态码	响应体
200 - OK	`PersistentVolumeClaim` schema
401 - Unauthorized	Empty

HTTP 方法

`PUT`

描述

replace the specified PersistentVolumeClaim

查询参数

参数	类型	描述
`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

参数	类型	描述
`fieldValidation`	`string`	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.

请求体参数

参数	类型	描述
`body`	`PersistentVolumeClaim` schema	`application/json` formatted

HTTP 响应

HTTP 状态码	响应体
200 - OK	`PersistentVolumeClaim` schema
201 - Created	`PersistentVolumeClaim` schema
401 - Unauthorized	Empty

/kubernetes/{cluster}/api/v1/namespaces/{namespace}/persistentvolumeclaims/{name}/status

HTTP 方法

`GET`

描述

read status of the specified PersistentVolumeClaim

HTTP 响应

HTTP 状态码	响应体
200 - OK	`PersistentVolumeClaim` schema
401 - Unauthorized	Empty

HTTP 方法

`PATCH`

描述

partially update status of the specified PersistentVolumeClaim

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	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.

参数	类型	描述
		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 响应

HTTP 状态码	响应体
200 - OK	`PersistentVolumeClaim` schema
401 - Unauthorized	Empty

HTTP 方法

`PUT`

描述

replace status of the specified PersistentVolumeClaim

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	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,

参数	类型	描述
		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.

请求体参数

参数	类型	描述
`body`	`PersistentVolumeClaim` schema	`application/json` formatted

HTTP 响应

HTTP 状态码	响应体
200 - OK	`PersistentVolumeClaim` schema
201 - Created	`PersistentVolumeClaim` schema
401 - Unauthorized	Empty

PersistentVolume [v1]

描述

PersistentVolume (PV) is a storage resource provisioned by an administrator. It is analogous to a node. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes

类型

`object`

规格

属性	类型	描述
`apiVersion`	`string`	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗
`kind`	`string`	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info:

属性	类型	描述
		https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`metadata`	`ObjectMeta`	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
`spec`	`object`	PersistentVolumeSpec is the specification of a persistent volume.
`status`	`object`	PersistentVolumeStatus is the current status of a persistent volume.

.spec

描述

PersistentVolumeSpec is the specification of a persistent volume.

类型

`object`

属性	类型	描述
`accessModes`	`array`	accessModes contains all ways the volume mounted. More info: https://kubernetes.io/docs/concepts/storage volumes#access-modes ↗
`awsElasticBlockStore`	`object`	Represents a Persistent Disk resource in A

属性	类型	描述
		An AWS EBS disk must exist before mount container. The disk must also be in the sam zone as the kubelet. An AWS EBS disk can mounted as read/write once. AWS EBS vol support ownership management and SELin relabeling.
`azureDisk`	`object`	AzureDisk represents an Azure Data Disk the host and bind mount to the pod.
`azureFile`	`object`	AzureFile represents an Azure File Service the host and bind mount to the pod.
`capacity`	`object`	capacity is the description of the persistent resources and capacity. More info: https://kubernetes.io/docs/concepts/storag volumes#capacity ↗
`cephfs`	`object`	Represents a Ceph Filesystem mount that lifetime of a pod Cephfs volumes do not su ownership management or SELinux relabe
`cinder`	`object`	Represents a cinder volume resource in O Cinder volume must exist before mounting container. The volume must also be in the region as the kubelet. Cinder volumes supp ownership management and SELinux relab

属性	类型	描述
`claimRef`	`object`	ObjectReference contains enough informa you inspect or modify the referred object.
`csi`	`object`	Represents storage that is managed by an CSI volume driver
`fc`	`object`	Represents a Fibre Channel volume. Fibre volumes can only be mounted as read/writ Fibre Channel volumes support ownership management and SELinux relabeling.
`flexVolume`	`object`	FlexPersistentVolumeSource represents a persistent volume resource that is provisioned/attached using an exec based
`flocker`	`object`	Represents a Flocker volume mounted by agent. One and only one of datasetName a datasetUUID should be set. Flocker volum support ownership management or SELinu relabeling.
`gcePersistentDisk`	`object`	Represents a Persistent Disk resource in C Compute Engine. A GCE PD must exist before mounting to a The disk must also be in the same GCE pr zone as the kubelet. A GCE PD can only b as read/write once or read-only many times

属性	类型	描述
		support ownership management and SELi relabeling.
`glusterfs`	`object`	Represents a Glusterfs mount that lasts th a pod. Glusterfs volumes do not support ov management or SELinux relabeling.
`hostPath`	`object`	Represents a host path mapped into a pod volumes do not support ownership manage SELinux relabeling.
`iscsi`	`object`	ISCSIPersistentVolumeSource represents disk. ISCSI volumes can only be mounted read/write once. ISCSI volumes support ov management and SELinux relabeling.
`local`	`object`	Local represents directly-attached storage affinity
`mountOptions`	`array`	mountOptions is the list of mount options, e "soft"]. Not validated - mount will simply fai invalid. More info: https://kubernetes.io/docs/concepts/storag volumes/#mount-options ↗
`nfs`	`object`	Represents an NFS mount that lasts the lif pod. NFS volumes do not support ownersh

属性	类型	描述
		management or SELinux relabeling.
`nodeAffinity`	`object`	VolumeNodeAffinity defines constraints tha nodes this volume can be accessed from.
`persistentVolumeReclaimPolicy`	`string`	persistentVolumeReclaimPolicy defines wh to a persistent volume when released from Valid options are Retain (default for manua PersistentVolumes), Delete (default for dyr provisioned PersistentVolumes), and Recy (deprecated). Recycle must be supported l volume plugin underlying this PersistentVo info: https://kubernetes.io/docs/concepts/storag volumes#reclaiming ↗ Possible enum values: • `"Delete"` means the volume will be d Kubernetes on release from its claim. T plugin must support Deletion. • `"Recycle"` means the volume will be back into the pool of unbound persisten on release from its claim. The volume p support Recycling. • `"Retain"` means the volume will be l current phase (Released) for manual re by the administrator. The default policy

属性	类型	描述
`photonPersistentDisk`	`object`	Represents a Photon Controller persistent resource.
`portworxVolume`	`object`	PortworxVolumeSource represents a Portv resource.
`quobyte`	`object`	Represents a Quobyte mount that lasts the a pod. Quobyte volumes do not support ow management or SELinux relabeling.
`rbd`	`object`	Represents a Rados Block Device mount t lifetime of a pod. RBD volumes support ow management and SELinux relabeling.
`scaleIO`	`object`	ScaleIOPersistentVolumeSource represen persistent ScaleIO volume
`storageClassName`	`string`	storageClassName is the name of Storage which this persistent volume belongs. Emp means that this volume does not belong to StorageClass.
`storageos`	`object`	Represents a StorageOS persistent volum

属性	类型	描述
`volumeAttributesClassName`	`string`	Name of VolumeAttributesClass to which tl persistent volume belongs. Empty value is allowed. When this field is not set, it indica volume does not belong to any VolumeAttr This field is mutable and can be changed b driver after a volume has been updated su a new class. For an unbound PersistentVo volumeAttributesClassName will be match unbound PersistentVolumeClaims during tl process. This is a beta field and requires e VolumeAttributesClass feature (off by defa
`volumeMode`	`string`	volumeMode defines if a volume is intende used with a formatted filesystem or to rema block state. Value of Filesystem is implied included in spec. Possible enum values: • `"Block"` means the volume will not b with a filesystem and will remain a raw device. • `"Filesystem"` means the volume will formatted with a filesystem.
`vsphereVolume`	`object`	Represents a vSphere volume resource.

.spec.accessModes

描述

accessModes contains all ways the volume can be mounted. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes

类型

`array`

.spec.accessModes[]

类型

`string`

.spec.awsElasticBlockStore

描述

Represents a Persistent Disk resource in AWS. An AWS EBS disk must exist before mounting to a container. The disk must also be in the same AWS zone as the kubelet. An AWS EBS disk can only be mounted as read/write once. AWS EBS volumes support ownership management and SELinux relabeling.

类型

`object`

必填

`volumeID`

属性	类型	描述
`fsType`	`string`	fsType is the filesystem type of the volume that you want to mou Ensure that the filesystem type is supported by the host operatir Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if ur More info: https://kubernetes.io/docs/concepts/storage/volumes#awselastic ↗

属性	类型	描述
`partition`	`integer`	partition is the partition in the volume that you want to mount. If the default is to mount by volume name. Examples: For volume you specify the partition as "1". Similarly, the volume partition for is "0" (or you can leave the property empty).
`readOnly`	`boolean`	readOnly value true will force the readOnly setting in VolumeMo info: https://kubernetes.io/docs/concepts/storage/volumes#awselastic ↗
`volumeID`	`string`	volumeID is unique ID of the persistent disk resource in AWS (A EBS volume). More info: https://kubernetes.io/docs/concepts/storage/volumes#awselastic ↗

.spec.azureDisk

描述

AzureDisk represents an Azure Data Disk mount on the host and bind mount to the pod.

类型

`object`

必填

`diskName` `diskURI`

属性	类型	描述
`cachingMode`	`string`	cachingMode is the Host Caching mode: None, Read Only, Read Write.

属性	类型	描述
		Possible enum values: - `"None"` - `"ReadOnly"` - `"ReadWrite"`
`diskName`	`string`	diskName is the Name of the data disk in the blob storage
`diskURI`	`string`	diskURI is the URI of data disk in the blob storage
`fsType`	`string`	fsType is Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
`kind`	`string`	kind expected values are Shared: multiple blob disks per storage account Dedicated: single blob disk per storage account Managed: azure managed data disk (only in managed availability set). defaults to shared Possible enum values: - `"Dedicated"` - `"Managed"` - `"Shared"`

属性	类型	描述
`readOnly`	`boolean`	readOnly Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

.spec.azureFile

描述

AzureFile represents an Azure File Service mount on the host and bind mount to the pod.

类型

`object`

必填

`secretName` `shareName`

属性	类型	描述
`readOnly`	`boolean`	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
`secretName`	`string`	secretName is the name of secret that contains Azure Storage Account Name and Key
`secretNamespace`	`string`	secretNamespace is the namespace of the secret that contains Azure Storage Account Name and Key default is the same as the Pod
`shareName`	`string`	shareName is the azure Share Name

.spec.capacity

描述

capacity is the description of the persistent volume's resources and capacity. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#capacity

类型

`object`

.spec.cephfs

描述

Represents a Ceph Filesystem mount that lasts the lifetime of a pod Cephfs volumes do not support ownership management or SELinux relabeling.

类型

`object`

必填

`monitors`

属性	类型	描述
`monitors`	`array`	monitors is Required: Monitors is a collection of Ceph monitors More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗
`path`	`string`	path is Optional: Used as the mounted root, rather than the full Ceph tree, default is /
`readOnly`	`boolean`	readOnly is Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts. More info:

属性	类型	描述
		https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗
`secretFile`	`string`	secretFile is Optional: SecretFile is the path to key ring for User, default is /etc/ceph/user.secret More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗
`secretRef`	`object`	SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace
`user`	`string`	user is Optional: User is the rados user name, default is admin More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗

.spec.cephfs.monitors

描述

monitors is Required: Monitors is a collection of Ceph monitors More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it

类型

`array`

.spec.cephfs.monitors[]

类型

`string`

.spec.cephfs.secretRef

描述

SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace

类型

`object`

属性	类型	描述
`name`	`string`	name is unique within a namespace to reference a secret resource.
`namespace`	`string`	namespace defines the space within which the secret name must be unique.

.spec.cinder

描述

Represents a cinder volume resource in Openstack. A Cinder volume must exist before mounting to a container. The volume must also be in the same region as the kubelet. Cinder volumes support ownership management and SELinux relabeling.

类型

`object`

必填

`volumeID`

属性	类型	描述
`fsType`	`string`	fsType Filesystem type to mount. Must be a filesystem type supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://examples.k8s.io/mysql-cinder-pd/README.md ↗
`readOnly`	`boolean`	readOnly is Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts. More info: https://examples.k8s.io/mysql-cinder-pd/README.md ↗
`secretRef`	`object`	SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace
`volumeID`	`string`	volumeID used to identify the volume in cinder. More info: https://examples.k8s.io/mysql-cinder-pd/README.md ↗

.spec.cinder.secretRef

描述

SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace

类型

`object`

属性	类型	描述
`name`	`string`	name is unique within a namespace to reference a secret resource.
`namespace`	`string`	namespace defines the space within which the secret name must be unique.

.spec.claimRef

描述

ObjectReference contains enough information to let you inspect or modify the referred object.

类型

`object`

属性	类型	描述
`apiVersion`	`string`	API version of the referent.
`fieldPath`	`string`	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 desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2

属性	类型	描述
		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: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.spec.csi

描述

Represents storage that is managed by an external CSI volume driver

类型

`object`

必填

`driver` `volumeHandle`

属性	类型	描述
`controllerExpandSecretRef`	`object`	SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace
`controllerPublishSecretRef`	`object`	SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace
`driver`	`string`	driver is the name of the driver to use for this volume. Required.
`fsType`	`string`	fsType to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs".
`nodeExpandSecretRef`	`object`	SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace

属性	类型	描述
`nodePublishSecretRef`	`object`	SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace
`nodeStageSecretRef`	`object`	SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace
`readOnly`	`boolean`	readOnly value to pass to ControllerPublishVolumeRequest. Defaults to false (read/write).
`volumeAttributes`	`object`	volumeAttributes of the volume to publish.
`volumeHandle`	`string`	volumeHandle is the unique volume name returned by the CSI volume plugin’s CreateVolume to refer to the volume on all subsequent calls. Required.

.spec.csi.controllerExpandSecretRef

描述

SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace

类型

`object`

属性	类型	描述
`name`	`string`	name is unique within a namespace to reference a secret resource.
`namespace`	`string`	namespace defines the space within which the secret name must be unique.

.spec.csi.controllerPublishSecretRef

描述

SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace

类型

`object`

属性	类型	描述
`name`	`string`	name is unique within a namespace to reference a secret resource.
`namespace`	`string`	namespace defines the space within which the secret name must be unique.

.spec.csi.nodeExpandSecretRef

描述

SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace

类型

`object`

属性	类型	描述
`name`	`string`	name is unique within a namespace to reference a secret resource.
`namespace`	`string`	namespace defines the space within which the secret name must be unique.

.spec.csi.nodePublishSecretRef

描述

SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace

类型

`object`

属性	类型	描述
`name`	`string`	name is unique within a namespace to reference a secret resource.
`namespace`	`string`	namespace defines the space within which the secret name must be unique.

.spec.csi.nodeStageSecretRef

描述

SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace

类型

`object`

属性	类型	描述
`name`	`string`	name is unique within a namespace to reference a secret resource.
`namespace`	`string`	namespace defines the space within which the secret name must be unique.

.spec.csi.volumeAttributes

描述

volumeAttributes of the volume to publish.

类型

`object`

.spec.fc

描述

Represents a Fibre Channel volume. Fibre Channel volumes can only be mounted as read/write once. Fibre Channel volumes support ownership management and SELinux relabeling.

类型

`object`

属性	类型	描述
`fsType`	`string`	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
`lun`	`integer`	lun is Optional: FC target lun number
`readOnly`	`boolean`	readOnly is Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
`targetWWNs`	`array`	targetWWNs is Optional: FC target worldwide names (WWNs)
`wwids`	`array`	wwids Optional: FC volume world wide identifiers (wwids) Either wwids or combination of targetWWNs and lun must be set, but not both simultaneously.

.spec.fc.targetWWNs

描述

targetWWNs is Optional: FC target worldwide names (WWNs)

类型

`array`

.spec.fc.targetWWNs[]

类型

`string`

.spec.fc.wwids

描述

wwids Optional: FC volume world wide identifiers (wwids) Either wwids or combination of targetWWNs and lun must be set, but not both simultaneously.

类型

`array`

.spec.fc.wwids[]

类型

`string`

.spec.flexVolume

描述

FlexPersistentVolumeSource represents a generic persistent volume resource that is provisioned/attached using an exec based plugin.

类型

`object`

必填

`driver`

属性	类型	描述
`driver`	`string`	driver is the name of the driver to use for this volume.

属性	类型	描述
`fsType`	`string`	fsType is the Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". The default filesystem depends on FlexVolume script.
`options`	`object`	options is Optional: this field holds extra command options if any.
`readOnly`	`boolean`	readOnly is Optional: defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
`secretRef`	`object`	SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace

.spec.flexVolume.options

描述

options is Optional: this field holds extra command options if any.

类型

`object`

.spec.flexVolume.secretRef

描述

SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace

类型

`object`

属性	类型	描述
`name`	`string`	name is unique within a namespace to reference a secret resource.
`namespace`	`string`	namespace defines the space within which the secret name must be unique.

.spec.flocker

描述

Represents a Flocker volume mounted by the Flocker agent. One and only one of datasetName and datasetUUID should be set. Flocker volumes do not support ownership management or SELinux relabeling.

类型

`object`

属性	类型	描述
`datasetName`	`string`	datasetName is Name of the dataset stored as metadata -> name on the dataset for Flocker should be considered as deprecated
`datasetUUID`	`string`	datasetUUID is the UUID of the dataset. This is unique identifier of a Flocker dataset

.spec.gcePersistentDisk

描述

Represents a Persistent Disk resource in Google Compute Engine. A GCE PD must exist before mounting to a container. The disk must also be in the same GCE project and zone as the kubelet. A GCE PD can only be mounted as read/write once or read-only many times. GCE PDs support ownership management and SELinux relabeling.

类型

`object`

必填

`pdName`

属性	类型	描述
`fsType`	`string`	fsType is filesystem type of the volume that you want to mount. Ensure that the filesystem type is supported by the host operatir system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "e unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersis ↗
`partition`	`integer`	partition is the partition in the volume that you want to mount. If omitted, the default is to mount by volume name. Examples: For volume /dev/sda1, you specify the partition as "1". Similarly, the partition for /dev/sda is "0" (or you can leave the property empty info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersis ↗
`pdName`	`string`	pdName is unique name of the PD resource in GCE. Used to ide the disk in GCE. More info:

属性	类型	描述
		https://kubernetes.io/docs/concepts/storage/volumes#gcepersis ↗
`readOnly`	`boolean`	readOnly here will force the ReadOnly setting in VolumeMounts Defaults to false. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersis ↗

.spec.glusterfs

描述

Represents a Glusterfs mount that lasts the lifetime of a pod. Glusterfs volumes do not support ownership management or SELinux relabeling.

类型

`object`

必填

`endpoints` `path`

属性	类型	描述
`endpoints`	`string`	endpoints is the endpoint name that details Glusterfs to More info: https://examples.k8s.io/volumes/glusterfs/README.md a-pod ↗
`endpointsNamespace`	`string`	endpointsNamespace is the namespace that contains G endpoint. If this field is empty, the EndpointNamespace to the same namespace as the bound PVC. More info:

属性	类型	描述
		https://examples.k8s.io/volumes/glusterfs/README.md# a-pod ↗
`path`	`string`	path is the Glusterfs volume path. More info: https://examples.k8s.io/volumes/glusterfs/README.md# a-pod ↗
`readOnly`	`boolean`	readOnly here will force the Glusterfs volume to be mou with read-only permissions. Defaults to false. More info: https://examples.k8s.io/volumes/glusterfs/README.md# a-pod ↗

.spec.hostPath

描述

Represents a host path mapped into a pod. Host path volumes do not support ownership management or SELinux relabeling.

类型

`object`

必填

`path`

属性	类型	描述
`path`	`string`	path of the directory on the host. If the path is a symlink, it will follow the link to the real path. More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath ↗
`type`	`string`	type for HostPath Volume Defaults to "" More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath ↗ Possible enum values: • `""` For backwards compatible, leave it empty if unset • `"BlockDevice"` A block device must exist at the given path • `"CharDevice"` A character device must exist at the given path • `"Directory"` A directory must exist at the given path • `"DirectoryOrCreate"` If nothing exists at the given path, an empty directory will be created there as needed with file mode 0755, having the same group and ownership with Kubelet. • `"File"` A file must exist at the given path • `"FileOrCreate"` If nothing exists at the given path, an empty file will be created there as needed with file mode 0644, having the same group and ownership with Kubelet. • `"Socket"` A UNIX socket must exist at the given path

.spec.iscsi

描述

ISCSIPersistentVolumeSource represents an ISCSI disk. ISCSI volumes can only be mounted as read/write once. ISCSI volumes support ownership management and SELinux relabeling.

类型

`object`

必填

`targetPortal` `iqn` `lun`

属性	类型	描述
`chapAuthDiscovery`	`boolean`	chapAuthDiscovery defines whether support iSCSI Discovery CHAP authentication
`chapAuthSession`	`boolean`	chapAuthSession defines whether support iSCSI Sessior CHAP authentication
`fsType`	`string`	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supporte by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#isc ↗
`initiatorName`	`string`	initiatorName is the custom iSCSI Initiator Name. If initiatorName is specified with iscsiInterface simultaneously, new iSCSI interface : will be created for the connection.

属性	类型	描述
`iqn`	`string`	iqn is Target iSCSI Qualified Name.
`iscsiInterface`	`string`	iscsiInterface is the interface Name that uses an iSCSI transport. Defaults to 'default' (tcp).
`lun`	`integer`	lun is iSCSI Target Lun number.
`portals`	`array`	portals is the iSCSI Target Portal List. The Portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).
`readOnly`	`boolean`	readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false.
`secretRef`	`object`	SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace
`targetPortal`	`string`	targetPortal is iSCSI Target Portal. The Portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).

.spec.iscsi.portals

描述

portals is the iSCSI Target Portal List. The Portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).

类型

`array`

.spec.iscsi.portals[]

类型

`string`

.spec.iscsi.secretRef

描述

SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace

类型

`object`

属性	类型	描述
`name`	`string`	name is unique within a namespace to reference a secret resource.
`namespace`	`string`	namespace defines the space within which the secret name must be unique.

.spec.local

描述

Local represents directly-attached storage with node affinity

类型

`object`

必填

`path`

属性	类型	描述
`fsType`	`string`	fsType is the filesystem type to mount. It applies only when the Path is a block device. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". The default value is to auto-select a filesystem if unspecified.
`path`	`string`	path of the full path to the volume on the node. It can be either a directory or block device (disk, partition, ...).

.spec.mountOptions

描述

mountOptions is the list of mount options, e.g. ["ro", "soft"]. Not validated - mount will simply fail if one is invalid. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes/#mount-options

类型

`array`

.spec.mountOptions[]

类型

`string`

.spec.nfs

描述

Represents an NFS mount that lasts the lifetime of a pod. NFS volumes do not support ownership management or SELinux relabeling.

类型

`object`

必填

`server` `path`

属性	类型	描述
`path`	`string`	path that is exported by the NFS server. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs ↗
`readOnly`	`boolean`	readOnly here will force the NFS export to be mounted with read-only permissions. Defaults to false. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs ↗
`server`	`string`	server is the hostname or IP address of the NFS server. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs ↗

.spec.nodeAffinity

描述

VolumeNodeAffinity defines constraints that limit what nodes this volume can be accessed from.

类型

`object`

属性	类型	描述
`required`	`object`	A node selector represents the union of the results of one or more label queries over a set of nodes; that is, it represents the OR of the selectors represented by the node selector terms.

.spec.nodeAffinity.required

描述

A node selector represents the union of the results of one or more label queries over a set of nodes; that is, it represents the OR of the selectors represented by the node selector terms.

类型

`object`

必填

`nodeSelectorTerms`

属性	类型	描述
`nodeSelectorTerms`	`array`	Required. A list of node selector terms. The terms are ORed.

.spec.nodeAffinity.required.nodeSelectorTerms

描述

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

类型

`array`

.spec.nodeAffinity.required.nodeSelectorTerms[]

描述

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.

类型

`object`

属性	类型	描述
`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.nodeAffinity.required.nodeSelectorTerms[].matchExpressions

描述

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

类型

`array`

.spec.nodeAffinity.required.nodeSelectorTerms[].matchExpressions[]

描述

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

类型

`object`

必填

`key` `operator`

属性	类型	描述
`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. Possible enum values: - `"DoesNotExist"` - `"Exists"` - `"Gt"` - `"In"` - `"Lt"` - `"NotIn"`
`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.nodeAffinity.required.nodeSelectorTerms[].matchExpressions[].values

描述

An array of string values. If the operator 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.

类型

`array`

.spec.nodeAffinity.required.nodeSelectorTerms[].matchExpressions[].values[]

类型

`string`

.spec.nodeAffinity.required.nodeSelectorTerms[].matchFields

描述

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

类型

`array`

.spec.nodeAffinity.required.nodeSelectorTerms[].matchFields[]

描述

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

类型

`object`

必填

`key` `operator`

属性	类型	描述
`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. Possible enum values: - `"DoesNotExist"` - `"Exists"` - `"Gt"` - `"In"` - `"Lt"` - `"NotIn"`
`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.nodeAffinity.required.nodeSelectorTerms[].matchFields[].values

描述

An array of string values. If the operator 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.

类型

`array`

.spec.nodeAffinity.required.nodeSelectorTerms[].matchFields[].values[]

类型

`string`

.spec.photonPersistentDisk

描述

Represents a Photon Controller persistent disk resource.

类型

`object`

必填

`pdID`

属性	类型	描述
`fsType`	`string`	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
`pdID`	`string`	pdID is the ID that identifies Photon Controller persistent disk

.spec.portworxVolume

描述

PortworxVolumeSource represents a Portworx volume resource.

类型

`object`

必填

`volumeID`

属性	类型	描述
`fsType`	`string`	fSType represents the filesystem type to mount Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs". Implicitly inferred to be "ext4" if unspecified.
`readOnly`	`boolean`	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

属性	类型	描述
`volumeID`	`string`	volumeID uniquely identifies a Portworx volume

.spec.quobyte

描述

Represents a Quobyte mount that lasts the lifetime of a pod. Quobyte volumes do not support ownership management or SELinux relabeling.

类型

`object`

必填

`registry` `volume`

属性	类型	描述
`group`	`string`	group to map volume access to Default is no group
`readOnly`	`boolean`	readOnly here will force the Quobyte volume to be mounted with read-only permissions. Defaults to false.
`registry`	`string`	registry represents a single or multiple Quobyte Registry services specified as a string as host:port pair (multiple entries are separated with commas) which acts as the central registry for volumes
`tenant`	`string`	tenant owning the given Quobyte volume in the Backend Used with dynamically provisioned Quobyte volumes,

属性	类型	描述
		value is set by the plugin
`user`	`string`	user to map volume access to Defaults to serivceaccount user
`volume`	`string`	volume is a string that references an already created Quobyte volume by name.

.spec.rbd

描述

Represents a Rados Block Device mount that lasts the lifetime of a pod. RBD volumes support ownership management and SELinux relabeling.

类型

`object`

必填

`monitors` `image`

属性	类型	描述
`fsType`	`string`	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#rbd ↗

属性	类型	描述
`image`	`string`	image is the rados image name. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it ↗
`keyring`	`string`	keyring is the path to key ring for RBDUser. Default is /etc/ceph/keyring. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it ↗
`monitors`	`array`	monitors is a collection of Ceph monitors. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it ↗
`pool`	`string`	pool is the rados pool name. Default is rbd. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it ↗
`readOnly`	`boolean`	readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it ↗
`secretRef`	`object`	SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace

属性	类型	描述
`user`	`string`	user is the rados user name. Default is admin. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it ↗

.spec.rbd.monitors

描述

monitors is a collection of Ceph monitors. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it

类型

`array`

.spec.rbd.monitors[]

类型

`string`

.spec.rbd.secretRef

描述

SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace

类型

`object`

属性	类型	描述
`name`	`string`	name is unique within a namespace to reference a secret resource.
`namespace`	`string`	namespace defines the space within which the secret name must be unique.

.spec.scaleIO

描述

ScaleIOPersistentVolumeSource represents a persistent ScaleIO volume

类型

`object`

必填

`gateway` `system` `secretRef`

属性	类型	描述
`fsType`	`string`	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Default is "xfs"
`gateway`	`string`	gateway is the host address of the ScaleIO API Gateway.

属性	类型	描述
`protectionDomain`	`string`	protectionDomain is the name of the ScaleIO Protection Domain for the configured storage.
`readOnly`	`boolean`	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
`secretRef`	`object`	SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace
`sslEnabled`	`boolean`	sslEnabled is the flag to enable/disable SSL communication with Gateway, default false
`storageMode`	`string`	storageMode indicates whether the storage for a volume should be ThickProvisioned or ThinProvisioned. Default is ThinProvisioned.
`storagePool`	`string`	storagePool is the ScaleIO Storage Pool associated with the protection domain.
`system`	`string`	system is the name of the storage system as configured in ScaleIO.

属性	类型	描述
`volumeName`	`string`	volumeName is the name of a volume already created in the ScaleIO system that is associated with this volume source.

.spec.scaleIO.secretRef

描述

SecretReference represents a Secret Reference. It has enough information to retrieve secret in any namespace

类型

`object`

属性	类型	描述
`name`	`string`	name is unique within a namespace to reference a secret resource.
`namespace`	`string`	namespace defines the space within which the secret name must be unique.

.spec.storageos

描述

Represents a StorageOS persistent volume resource.

类型

`object`

属性	类型	描述
`fsType`	`string`	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
`readOnly`	`boolean`	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
`secretRef`	`object`	ObjectReference contains enough information to let you inspect or modify the referred object.
`volumeName`	`string`	volumeName is the human-readable name of the StorageOS volume. Volume names are only unique within a namespace.
`volumeNamespace`	`string`	volumeNamespace specifies the scope of the volume within StorageOS. If no namespace is specified then the Pod's namespace will be used. This allows the Kubernetes name scoping to be mirrored within StorageOS for tighter integration. Set VolumeName to any name to override the default behaviour. Set to "default" if you are not using namespaces within StorageOS. Namespaces that do not pre-exist within StorageOS will be created.

.spec.storageos.secretRef

描述

ObjectReference contains enough information to let you inspect or modify the referred object.

类型

`object`

属性	类型	描述
`apiVersion`	`string`	API version of the referent.
`fieldPath`	`string`	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 desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is 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: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

属性	类型	描述
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.spec.vsphereVolume

描述

Represents a vSphere volume resource.

类型

`object`

必填

`volumePath`

属性	类型	描述
`fsType`	`string`	fsType is filesystem type to mount. Must be a filesystem type supported by the host operating

属性	类型	描述
		system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
`storagePolicyID`	`string`	storagePolicyID is the storage Policy Based Management (SPBM) profile ID associated with the StoragePolicyName.
`storagePolicyName`	`string`	storagePolicyName is the storage Policy Based Management (SPBM) profile name.
`volumePath`	`string`	volumePath is the path that identifies vSphere volume vmdk

.status

描述

PersistentVolumeStatus is the current status of a persistent volume.

类型

`object`

属性	类型	描述
`lastPhaseTransitionTime`	`string`	Time is a wrapper around time.Time which suppoı correct marshaling to YAML and JSON. Wrappers provided for many of the factory methods that the package offers.

属性	类型	描述
`message`	`string`	message is a human-readable message indicating details about why the volume is in this state.
`phase`	`string`	phase indicates if a volume is available, bound to a claim, or released by a claim. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#phase Possible enum values: • `"Available"` used for PersistentVolumes that are not yet bound Available volumes are held by the binder and matched to PersistentVolumeClaims • `"Bound"` used for PersistentVolumes that are bound • `"Failed"` used for PersistentVolumes that failed to be correctly recycled or deleted after being released from a claim • `"Pending"` used for PersistentVolumes that are not available • `"Released"` used for PersistentVolumes where the bound PersistentVolumeClaim was deleted released volumes must be recycled before becoming available again this phase is used by persistent volume claim binder to signal to another process to reclaim the resource

属性	类型	描述
`reason`	`string`	reason is a brief CamelCase string that describes failure and is meant for machine parsing and tidy display in the CLI.

API 端点

可用的 API 端点如下：

- `/kubernetes/{cluster}/api/v1/persistentvolumes`
  - `DELETE`: delete collection of PersistentVolume
  - `GET`: list objects of kind PersistentVolume
  - `POST`: create a new PersistentVolume
- `/kubernetes/{cluster}/api/v1/persistentvolumes/{name}`
  - `DELETE`: delete the specified PersistentVolume
  - `GET`: read the specified PersistentVolume
  - `PATCH`: partially update the specified PersistentVolume
  - `PUT`: replace the specified PersistentVolume
- `/kubernetes/{cluster}/api/v1/persistentvolumes/{name}/status`
  - `GET`: read status of the specified PersistentVolume
  - `PATCH`: partially update status of the specified PersistentVolume
  - `PUT`: replace status of the specified PersistentVolume

/kubernetes/{cluster}/api/v1/persistentvolumes

HTTP 方法

`DELETE`

描述

delete collection of PersistentVolume

HTTP 响应

HTTP 状态码	响应体
200 - OK	`Status` schema
401 - Unauthorized	Empty

HTTP 方法

`GET`

描述

list objects of kind PersistentVolume

HTTP 响应

HTTP 状态码	响应体
200 - OK	`PersistentVolumeList` schema
401 - Unauthorized	Empty

HTTP 方法

`POST`

描述

create a new PersistentVolume

查询参数

参数	类型	描述
`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

参数	类型	描述
`fieldValidation`	`string`	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.

请求体参数

参数	类型	描述
`body`	`PersistentVolume` schema	`application/json` formatted

HTTP 响应

HTTP 状态码	响应体
200 - OK	`PersistentVolume` schema
201 - Created	`PersistentVolume` schema
202 - Accepted	`PersistentVolume` schema
401 - Unauthorized	Empty

/kubernetes/{cluster}/api/v1/persistentvolumes/{name}

HTTP 方法

`DELETE`

描述

delete the specified PersistentVolume

查询参数

参数	类型	描述
`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 响应

HTTP 状态码	响应体
200 - OK	`Status` schema
202 - Accepted	`Status` schema
401 - Unauthorized	Empty

HTTP 方法

`GET`

描述

read the specified PersistentVolume

HTTP 响应

HTTP 状态码	响应体
200 - OK	`PersistentVolume` schema
401 - Unauthorized	Empty

HTTP 方法

`PATCH`

描述

partially update the specified PersistentVolume

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	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.

HTTP 响应

HTTP 状态码	响应体
200 - OK	`PersistentVolume` schema
401 - Unauthorized	Empty

HTTP 方法

`PUT`

描述

replace the specified PersistentVolume

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	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.

请求体参数

参数	类型	描述
`body`	`PersistentVolume` schema	`application/json` formatted

HTTP 响应

HTTP 状态码	响应体
200 - OK	`PersistentVolume` schema
201 - Created	`PersistentVolume` schema
401 - Unauthorized	Empty

/kubernetes/{cluster}/api/v1/persistentvolumes/{name}/status

HTTP 方法

`GET`

描述

read status of the specified PersistentVolume

HTTP 响应

HTTP 状态码	响应体
200 - OK	`PersistentVolume` schema
401 - Unauthorized	Empty

HTTP 方法

`PATCH`

描述

partially update status of the specified PersistentVolume

查询参数

参数	类型	描述
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun

参数	类型	描述
		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
`fieldValidation`	`string`	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.

HTTP 响应

HTTP 状态码	响应体
200 - OK	`PersistentVolume` schema
401 - Unauthorized	Empty

HTTP 方法

`PUT`

描述

replace status of the specified PersistentVolume

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	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.

请求体参数

参数	类型	描述
`body`	`PersistentVolume` schema	`application/json` formatted

HTTP 响应

HTTP 状态码	响应体
200 - OK	`PersistentVolume` schema
201 - Created	`PersistentVolume` schema

HTTP 状态码	响应体
401 - Unauthorized	Empty

StorageClass [storage.k8s.io/v1]

描述

StorageClass describes the parameters for a class of storage for which PersistentVolumes can be dynamically provisioned. StorageClasses are non-namespaced; the name of the storage class according to etcd is in ObjectMeta.Name.

类型

`object`

必填

`provisioner`

规格

属性	类型	描述
`allowVolumeExpansion`	`boolean`	allowVolumeExpansion shows whether the storage class allow volume expand.
`allowedTopologies`	`array`	allowedTopologies restrict the node topologies where volumes can be dynamically provisioned. Each volume plugin defines its own supported topology specifications. An empty TopologySelectorTerm list means there is no

属性	类型	描述
		topology restriction. This field is only honored by servers that enable the VolumeScheduling feature.
`apiVersion`	`string`	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗
`kind`	`string`	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`metadata`	`ObjectMeta`	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
`mountOptions`	`array`	mountOptions controls the mountOptions for dynamically provisioned PersistentVolumes of this storage class. e.g. ["ro", "soft"]. Not validated - mount of the PVs will simply fail if one is invalid.

属性	类型	描述
`parameters`	`object`	parameters holds the parameters for the provisioner that should create volumes of this storage class.
`provisioner`	`string`	provisioner indicates the type of the provisioner.
`reclaimPolicy`	`string`	reclaimPolicy controls the reclaimPolicy for dynamically provisioned PersistentVolumes of this storage class. Defaults to Delete. Possible enum values: • `"Delete"` means the volume will be deleted from Kubernetes on release from its claim. The volume plugin must support Deletion. • `"Recycle"` means the volume will be recycled back into the pool of unbound persistent volumes on release from its claim. The volume plugin must support Recycling. • `"Retain"` means the volume will be left in its current phase (Released) for manual reclamation by the administrator. The default policy is Retain.
`volumeBindingMode`	`string`	volumeBindingMode indicates how PersistentVolumeClaims should be provisioned and bound. When unset, VolumeBindingImmediate is used. This field is only honored by servers that enable the VolumeScheduling feature.

属性	类型	描述
		Possible enum values: - `"Immediate"` indicates that PersistentVolumeClaims should be immediately provisioned and bound. This is the default mode. - `"WaitForFirstConsumer"` indicates that PersistentVolumeClaims should not be provisioned and bound until the first Pod is created that references the PeristentVolumeClaim. The volume provisioning and binding will occur during Pod scheduing.

.allowedTopologies

描述

allowedTopologies restrict the node topologies where volumes can be dynamically provisioned. Each volume plugin defines its own supported topology specifications. An empty TopologySelectorTerm list means there is no topology restriction. This field is only honored by servers that enable the VolumeScheduling feature.

类型

`array`

.allowedTopologies[]

描述

A topology selector term represents the result of label queries. A null or empty topology selector term matches no objects. The requirements of them are ANDed. It provides a subset of functionality as NodeSelectorTerm. This is an alpha feature and may change in the future.

类型

`object`

属性	类型	描述
`matchLabelExpressions`	`array`	A list of topology selector requirements by labels.

.allowedTopologies[].matchLabelExpressions

描述

A list of topology selector requirements by labels.

类型

`array`

.allowedTopologies[].matchLabelExpressions[]

描述

A topology selector requirement is a selector that matches given label. This is an alpha feature and may change in the future.

类型

`object`

必填

`key` `values`

属性	类型	描述
`key`	`string`	The label key that the selector applies to.

属性	类型	描述
`values`	`array`	An array of string values. One value must match the label to be selected. Each entry in Values is ORed.

.allowedTopologies[].matchLabelExpressions[].values

描述

An array of string values. One value must match the label to be selected. Each entry in Values is ORed.

类型

`array`

.allowedTopologies[].matchLabelExpressions[].values[]

类型

`string`

.mountOptions

描述

mountOptions controls the mountOptions for dynamically provisioned PersistentVolumes of this storage class. e.g. ["ro", "soft"]. Not validated - mount of the PVs will simply fail if one is invalid.

类型

`array`

.mountOptions[]

类型

`string`

.parameters

描述

parameters holds the parameters for the provisioner that should create volumes of this storage class.

类型

`object`

API 端点

可用的 API 端点如下：

- `/kubernetes/{cluster}/apis/storage.k8s.io/v1/storageclasses`
  - `DELETE`: delete collection of StorageClass
  - `GET`: list objects of kind StorageClass
  - `POST`: create a new StorageClass
- `/kubernetes/{cluster}/apis/storage.k8s.io/v1/storageclasses/{name}`
  - `DELETE`: delete the specified StorageClass
  - `GET`: read the specified StorageClass
  - `PATCH`: partially update the specified StorageClass
  - `PUT`: replace the specified StorageClass

/kubernetes/{cluster}/apis/storage.k8s.io/v1/storageclasses

HTTP 方法

`DELETE`

描述

delete collection of StorageClass

HTTP 响应

HTTP 状态码	响应体
200 - OK	`Status` schema
401 - Unauthorized	Empty

HTTP 方法

`GET`

描述

list objects of kind StorageClass

HTTP 响应

HTTP 状态码	响应体
200 - OK	`StorageClassList` schema
401 - Unauthorized	Empty

HTTP 方法

`POST`

描述

create a new StorageClass

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore:

参数	类型	描述
		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.

请求体参数

参数	类型	描述
`body`	`StorageClass` schema	`application/json` formatted

HTTP 响应

HTTP 状态码	响应体
200 - OK	`StorageClass` schema
201 - Created	`StorageClass` schema
202 - Accepted	`StorageClass` schema
401 - Unauthorized	Empty

/kubernetes/{cluster}/apis/storage.k8s.io/v1/storageclasses/{name}

HTTP 方法

`DELETE`

描述

delete the specified StorageClass

查询参数

参数	类型	描述
`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 响应

HTTP 状态码	响应体
200 - OK	`Status` schema
202 - Accepted	`Status` schema
401 - Unauthorized	Empty

HTTP 方法

`GET`

描述

read the specified StorageClass

HTTP 响应

HTTP 状态码	响应体
200 - OK	`StorageClass` schema
401 - Unauthorized	Empty

HTTP 方法

`PATCH`

描述

partially update the specified StorageClass

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	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.

HTTP 响应

HTTP 状态码	响应体
200 - OK	`StorageClass` schema
401 - Unauthorized	Empty

HTTP 方法

`PUT`

描述

replace the specified StorageClass

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	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.

请求体参数

参数	类型	描述
`body`	`StorageClass` schema	`application/json` formatted

HTTP 响应

HTTP 状态码	响应体
200 - OK	`StorageClass` schema
201 - Created	`StorageClass` schema
401 - Unauthorized	Empty

Ceph Storage APIs

CephCluster [cephclusters.cep CephFilesystem [cephfilesyste CephBlockI

CephObjectStore [cephobjectst CephObjectStoreUser [cephobjectstoreusers.

CephCluster [cephclusters.ceph.rook.io/v1]

描述

CephCluster is a Ceph storage cluster

类型

`object`

必填

`metadata` `spec`

规格

属性	类型	描述
`apiVersion`	`string`	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗
`kind`	`string`	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In

属性	类型	描述
		CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`metadata`	`ObjectMeta`	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
`spec`	`object`	ClusterSpec represents the specification of Ceph Cluster
`status`	`object`	ClusterStatus represents the status of a Ceph cluster

.spec

描述

ClusterSpec represents the specification of Ceph Cluster

类型

`object`

属性	类型	描述
`annotations`	`object`	The annotations-related co each Pod related object.
`cephConfig`	`object`	Ceph Config options

属性	类型	描述
`cephConfigFromSecret`	`object`	CephConfigFromSecret wo CephConfig but takes confi reference.
`cephVersion`	`object`	The version information tha orchestrate a particular ver
`cleanupPolicy`	`object`	Indicates user intent when orchestration and should n deletion is not imminent.
`continueUpgradeAfterChecksEvenIfNotHealthy`	`boolean`	ContinueUpgradeAfterChe defines if an upgrade shoul are not clean
`crashCollector`	`object`	A spec for the crash contro
`csi`	`object`	CSI Driver Options applied
`dashboard`	`object`	Dashboard settings
`dataDirHostPath`	`string`	The path on the host where persisted

属性	类型	描述
`disruptionManagement`	`object`	A spec for configuring disru
`external`	`object`	Whether the Ceph Cluster i Kubernetes cluster mon, m discover daemons will not b clusters.
`healthCheck`	`object`	Internal daemon healthche
`labels`	`object`	The labels-related configura Pod related object.
`logCollector`	`object`	Logging represents logging
`mgr`	`object`	A spec for mgr related optic
`mon`	`object`	A spec for mon related opti
`monitoring`	`object`	Prometheus based Monitor
`network`	`object`	Network related configurati

属性	类型	描述
`placement`	`object`	
`priorityClassNames`	`object`	PriorityClassNames sets pr components
`removeOSDsIfOutAndSafeToRemove`	`boolean`	Remove the OSD that is ou only if this option is true
`resources`	`object`	Resources set resource rec
`security`	`object`	Security represents securit
`skipUpgradeChecks`	`boolean`	SkipUpgradeChecks define be forced even if one of the
`storage`	`object`	A spec for available storage it should be used
`upgradeOSDRequiresHealthyPGs`	`boolean`	UpgradeOSDRequiresHeal upgrade requires PGs are OSD upgrade process won healthy. This configuration `skipUpgradeChecks` is `t`

属性	类型	描述
`waitTimeoutForHealthyOSDInMinutes`	`integer`	WaitTimeoutForHealthyOS time the operator would wa stopped for upgrade or rest exceeds and OSD is not ok operator would skip upgrad and proceed with the next c `continueUpgradeAfterCh` is `false`. If `continueUpgradeAfterCh` is `true`, then operator wo upgrade of an OSD even if the timeout. This timeout w `skipUpgradeChecks` is `t` timeout is 10 minutes.

.spec.annotations

描述

The annotations-related configuration to add/set on each Pod related object.

类型

`object`

.spec.cephConfig

描述

Ceph Config options

类型

`object`

.spec.cephConfigFromSecret

描述

CephConfigFromSecret works exactly like CephConfig but takes config value from Secret Key reference.

类型

`object`

.spec.cephVersion

描述

The version information that instructs Rook to orchestrate a particular version of Ceph.

类型

`object`

属性	类型	描述
`allowUnsupported`	`boolean`	Whether to allow unsupported versions (do not set to true in production)
`image`	`string`	Image is the container image used to launch the ceph daemons, such as quay.io/ceph/ceph: The full list of images can be found at https://quay.io/repository/ceph/ceph?tab=tags ↗
`imagePullPolicy`	`string`	ImagePullPolicy describes a policy for if/when to pull a container image One of Always, Never, IfNotPresent.

.spec.cleanupPolicy

描述

Indicates user intent when deleting a cluster; blocks orchestration and should not be set if cluster deletion is not imminent.

类型

`object`

属性	类型	描述
`allowUninstallWithVolumes`	`boolean`	AllowUninstallWithVolumes defines whether we can proceed with the uninstall if they are RBD images still present
`confirmation`	`string`	Confirmation represents the cleanup confirmation
`sanitizeDisks`	`object`	SanitizeDisks represents way we sanitize disks
`wipeDevicesFromOtherClusters`	`boolean`	WipeDevicesFromOtherClusters wipes the OSD disks belonging to other clusters. This is useful in scenarios where ceph cluster was reinstalled but OSD disk still contains the metadata from previous ceph cluster.

.spec.cleanupPolicy.sanitizeDisks

描述

SanitizeDisks represents way we sanitize disks

类型

`object`

属性	类型	描述
`dataSource`	`string`	DataSource is the data source to use to sanitize the disk with
`iteration`	`integer`	Iteration is the number of pass to apply the sanitizing
`method`	`string`	Method is the method we use to sanitize disks

.spec.crashCollector

描述

A spec for the crash controller

类型

`object`

属性	类型	描述
`daysToRetain`	`integer`	DaysToRetain represents the number of days to retain crash until they get pruned
`disable`	`boolean`	Disable determines whether we should enable the crash collector

.spec.csi

描述

CSI Driver Options applied per cluster.

类型

`object`

属性	类型	描述
`cephfs`	`object`	CephFS defines CSI Driver settings for CephFS driver.
`readAffinity`	`object`	ReadAffinity defines the read affinity settings for CSI driver.
`skipUserCreation`	`boolean`	SkipUserCreation determines whether CSI users and their associated secrets should be skipped. If set to true, the user must manually manage these secrets.

.spec.csi.cephfs

描述

CephFS defines CSI Driver settings for CephFS driver.

类型

`object`

属性	类型	描述
`fuseMountOptions`	`string`	FuseMountOptions defines the mount options for ceph fuse mounter.
`kernelMountOptions`	`string`	KernelMountOptions defines the mount options for kernel mounter.

.spec.csi.readAffinity

描述

ReadAffinity defines the read affinity settings for CSI driver.

类型

`object`

属性	类型	描述
`crushLocationLabels`	`array`	CrushLocationLabels defines which node labels to use as CRUSH location. This should correspond to the values set in the CRUSH map.
`enabled`	`boolean`	Enables read affinity for CSI driver.

.spec.csi.readAffinity.crushLocationLabels

描述

CrushLocationLabels defines which node labels to use as CRUSH location. This should correspond to the values set in the CRUSH map.

类型

`array`

.spec.csi.readAffinity.crushLocationLabels[]

类型

`string`

.spec.dashboard

描述

Dashboard settings

类型

`object`

属性	类型	描述
`enabled`	`boolean`	Enabled determines whether to enable the dashboard
`port`	`integer`	Port is the dashboard webserver port
`prometheusEndpoint`	`string`	Endpoint for the Prometheus host
`prometheusEndpointSSLVerify`	`boolean`	Whether to verify the ssl endpoint for prometheus. Set to false for a self-signed cert.

属性	类型	描述
`ssl`	`boolean`	SSL determines whether SSL should be used
`urlPrefix`	`string`	URLPrefix is a prefix for all URLs to use the dashboard with a reverse proxy

.spec.disruptionManagement

描述

A spec for configuring disruption management.

类型

`object`

属性	类型	描述
`machineDisruptionBudgetNamespace`	`string`	Deprecated. Namespace to look for MD
`manageMachineDisruptionBudgets`	`boolean`	Deprecated. This enables managemen
`managePodBudgets`	`boolean`	This enables management of poddisrup
`osdMaintenanceTimeout`	`integer`	OSDMaintenanceTimeout sets how ma for drained failure domains it only works minutes

属性	类型	描述
`pgHealthCheckTimeout`	`integer`	DEPRECATED: PGHealthCheckTimeo
`pgHealthyRegex`	`string`	PgHealthyRegex is the regular express should be considered healthy. The defa `^(active\+clean|active\+clean\+s`

.spec.external

描述

Whether the Ceph Cluster is running external to this Kubernetes cluster mon, mgr, osd, mds, and discover daemons will not be created for external clusters.

类型

`object`

属性	类型	描述
`enable`	`boolean`	Enable determines whether external mode is enabled or not

.spec.healthCheck

描述

Internal daemon healthchecks and liveness probe

类型

`object`

属性	类型	描述
`daemonHealth`	`object`	DaemonHealth is the health check for a given daemon
`livenessProbe`	`object`	LivenessProbe allows changing the livenessProbe configuration for a given daemon
`startupProbe`	`object`	StartupProbe allows changing the startupProbe configuration for a given daemon

.spec.healthCheck.daemonHealth

描述

DaemonHealth is the health check for a given daemon

类型

`object`

属性	类型	描述
`mon`	`object`	Monitor represents the health check settings for the Ceph monitor
`osd`	`object`	ObjectStorageDaemon represents the health check settings for the Ceph OSDs
`status`	`object`	Status represents the health check settings for the Ceph health

.spec.healthCheck.daemonHealth.mon

描述

Monitor represents the health check settings for the Ceph monitor

类型

`object`

属性	类型	描述
`disabled`	`boolean`	
`interval`	`string`	Interval is the internal in second or minute for the health check to run like 60s for 60 seconds
`timeout`	`string`	

.spec.healthCheck.daemonHealth.osd

描述

ObjectStorageDaemon represents the health check settings for the Ceph OSDs

类型

`object`

属性	类型	描述
`disabled`	`boolean`	
`interval`	`string`	Interval is the internal in second or minute for the health check to run like 60s for 60 seconds
`timeout`	`string`	

.spec.healthCheck.daemonHealth.status

描述

Status represents the health check settings for the Ceph health

类型

`object`

属性	类型	描述
`disabled`	`boolean`	
`interval`	`string`	Interval is the internal in second or minute for the health check to run like 60s for 60 seconds
`timeout`	`string`	

.spec.healthCheck.livenessProbe

描述

LivenessProbe allows changing the livenessProbe configuration for a given daemon

类型

`object`

.spec.healthCheck.startupProbe

描述

StartupProbe allows changing the startupProbe configuration for a given daemon

类型

`object`

.spec.labels

描述

The labels-related configuration to add/set on each Pod related object.

类型

`object`

.spec.logCollector

描述

Logging represents loggings settings

类型

`object`

属性	类型	描述
`enabled`	`boolean`	Enabled represents whether the log collector is enabled
`maxLogSize`	`integer|string`	MaxLogSize is the maximum size of the log per ceph daemons. Must be at least 1M.
`periodicity`	`string`	Periodicity is the periodicity of the log rotation.

.spec.mgr

描述

A spec for mgr related options

类型

`object`

属性	类型	描述
`allowMultiplePerNode`	`boolean`	AllowMultiplePerNode allows to run multiple managers on the same node (not recommended)
`count`	`integer`	Count is the number of manager daemons to run
`hostNetwork`	`boolean`	Whether host networking is enabled for the Ceph Mgr. If not set, the network settings from CephCluster.spec.networking will be applied.
`modules`	`array`	Modules is the list of ceph manager modules to enable/disable

.spec.mgr.modules

描述

Modules is the list of ceph manager modules to enable/disable

类型

`array`

.spec.mgr.modules[]

描述

Module represents mgr modules that the user wants to enable or disable

类型

`object`

属性	类型	描述
`enabled`	`boolean`	Enabled determines whether a module should be enabled or not
`name`	`string`	Name is the name of the ceph manager module
`settings`	`object`	Settings to further configure the module

.spec.mgr.modules[].settings

描述

Settings to further configure the module

类型

`object`

属性	类型	描述
`balancerMode`	`string`	BalancerMode sets the `balancer` module with different modes like `upmap`, `crush-compact` etc

.spec.mon

描述

A spec for mon related options

类型

`object`

属性	类型	描述
`allowMultiplePerNode`	`boolean`	AllowMultiplePerNode determines if we can run multiple monitors on the same node (not recommended)
`count`	`integer`	Count is the number of Ceph monitors
`externalMonIDs`	`array`	ExternalMonIDs - optional list of monitor IDs which are deployed externally and not managed by Rook. If set, Rook will not remove mons with given IDs from quorum. This parameter is used only for local Rook cluster running in normal mode and will be ignored if external or stretched mode is used. leading
`failureDomainLabel`	`string`	
`stretchCluster`	`object`	StretchCluster is the stretch cluster specification
`volumeClaimTemplate`	`object`	VolumeClaimTemplate is the PVC definition
`zones`	`array`	Zones are specified when we want to provide zonal awareness to mons

.spec.mon.externalMonIDs

描述

ExternalMonIDs - optional list of monitor IDs which are deployed externally and not managed by Rook. If set, Rook will not remove mons with given IDs from quorum. This parameter is used only for local Rook cluster running in normal mode and will be ignored if external or stretched mode is used. leading

类型

`array`

.spec.mon.externalMonIDs[]

类型

`string`

.spec.mon.stretchCluster

描述

StretchCluster is the stretch cluster specification

类型

`object`

属性	类型	描述
`failureDomainLabel`	`string`	FailureDomainLabel the failure domain name (e,g: zone)
`subFailureDomain`	`string`	SubFailureDomain is the failure domain within a zone
`zones`	`array`	Zones is the list of zones

.spec.mon.stretchCluster.zones

描述

Zones is the list of zones

类型

`array`

.spec.mon.stretchCluster.zones[]

描述

MonZoneSpec represents the specification of a zone in a Ceph Cluster

类型

`object`

属性	类型	描述
`arbiter`	`boolean`	Arbiter determines if the zone contains the arbiter used for stretch cluster mode
`name`	`string`	Name is the name of the zone
`volumeClaimTemplate`	`object`	VolumeClaimTemplate is the PVC template

.spec.mon.stretchCluster.zones[].volumeClaimTemplate

描述

VolumeClaimTemplate is the PVC template

类型

`object`

属性	类型	描述
`metadata`	`ObjectMeta`	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata ↗
`spec`	`object`	spec defines the desired characteristics of a volume requested by a pod author. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims ↗

.spec.mon.stretchCluster.zones[].volumeClaimTemplate.spec

描述

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

类型

`object`

属性	类型	描述
`accessModes`	`array`	accessModes contains the desired access mod volume should have. More info: https://kubernetes.io/docs/concepts/storage/per volumes#access-modes-1 ↗
`dataSource`	`object`	dataSource field can be used to specify either:

属性	类型	描述
		• An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) • An existing PVC (PersistentVolumeClaim) If provisioner or an external controller can sup the specified data source, it will create a nev volume based on the contents of the specifie source. When the AnyVolumeDataSource fe gate is enabled, dataSource contents will be to dataSourceRef, and dataSourceRef conte be copied to dataSource when dataSourceRef.namespace is not specified. namespace is specified, then dataSourceRe not be copied to dataSource.
`dataSourceRef`	`object`	dataSourceRef specifies the object from which t populate the volume with data, if a non-empty v is desired. This may be any object from a non-e API group (non core object) or a PersistentVolumeClaim object. When this field i specified, volume binding will only succeed if th of the specified object matches some installed v populator or dynamic provisioner. This field will the functionality of the dataSource field and as s both fields are non-empty, they must have the s value. For backwards compatibility, when name isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to t same value automatically if one of them is empt the other is non-empty. When namespace is sp in dataSourceRef, dataSource isn't set to the sa value and must be empty. There are three impo differences between dataSource and dataSourc

属性	类型	描述
		• While dataSource only allows two specific ty objects, dataSourceRef allows any non-core as well as PersistentVolumeClaim objects. • While dataSource ignores disallowed values (dropping them), dataSourceRef preserves a values, and generates an error if a disallowe value is specified. • While dataSource only allows local objects, dataSourceRef allows objects in any names (Beta) Using this field requires the AnyVolumeDataSource feature gate to be en (Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource featur to be enabled.
`resources`	`object`	resources represents the minimum resources th volume should have. Users are allowed to spec resource requirements that are lower than previ value but must still be higher than capacity reco the status field of the claim. More info: https://kubernetes.io/docs/concepts/storage/per volumes#resources ↗
`selector`	`object`	selector is a label query over volumes to consid binding.

属性	类型	描述
`storageClassName`	`string`	storageClassName is the name of the StorageC required by the claim. More info: https://kubernetes.io/docs/concepts/storage/per volumes#class-1 ↗
`volumeAttributesClassName`	`string`	volumeAttributesClassName may be used to se VolumeAttributesClass used by this claim. If spe the CSI driver will create or update the volume v attributes defined in the corresponding VolumeAttributesClass. This has a different pur than storageClassName, it can be changed afte claim is created. An empty string or nil value inc that no VolumeAttributesClass will be applied to claim. If the claim enters an Infeasible error stat field can be reset to its previous value (including cancel the modification. If the resource referred volumeAttributesClass does not exist, this PersistentVolumeClaim will be set to a Pending as reflected by the modifyVolumeStatus field, u such as a resource exists. More info: https://kubernetes.io/docs/concepts/storage/vol attributes-classes/ ↗
`volumeMode`	`string`	volumeMode defines what type of volume is req by the claim. Value of Filesystem is implied whe included in claim spec.
`volumeName`	`string`	volumeName is the binding reference to the PersistentVolume backing this claim.

.spec.mon.stretchCluster.zones[].volumeClaimTemplate.spec.accessModes

描述

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

类型

`array`

.spec.mon.stretchCluster.zones[].volumeClaimTemplate.spec.accessModes[]

类型

`string`

.spec.mon.stretchCluster.zones[].volumeClaimTemplate.spec.dataSource

描述

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

类型

`object`

必填

`kind` `name`

属性	类型	描述
`apiGroup`	`string`	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
`kind`	`string`	Kind is the type of resource being referenced
`name`	`string`	Name is the name of resource being referenced

.spec.mon.stretchCluster.zones[].volumeClaimTemplate.spec.dataSourceRef

描述

dataSourceRef specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a PersistentVolumeClaim object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replace the functionality of the dataSource field and as such if both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in dataSourceRef, dataSource isn't set to the same value and must be empty. There are three important differences between dataSource and dataSourceRef: * While dataSource only allows two specific types of objects, dataSourceRef allows any non-core object, as well as PersistentVolumeClaim objects. * While dataSource ignores disallowed values (dropping them), dataSourceRef preserves all values, and generates an error if a disallowed value is specified. * While dataSource only allows local objects, dataSourceRef allows objects in any namespaces. (Beta) Using this field requires the AnyVolumeDataSource feature gate to be enabled.

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

类型

`object`

必填

`kind` `name`

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

.spec.mon.stretchCluster.zones[].volumeClaimTemplate.spec.resources

描述

resources represents the minimum resources the volume should have. Users are allowed to specify resource requirements that are lower than previous value but must still be higher than capacity recorded in the status field of the claim. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources

类型

`object`

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

.spec.mon.stretchCluster.zones[].volumeClaimTemplate.spec.resources.limits

描述

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

类型

`object`

.spec.mon.stretchCluster.zones[].volumeClaimTemplate.spec.resources.requests

描述

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/

类型

`object`

.spec.mon.stretchCluster.zones[].volumeClaimTemplate.spec.selector

描述

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

类型

`object`

属性	类型	描述
`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.mon.stretchCluster.zones[].volumeClaimTemplate.spec.selector.matchExpressions

描述

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

类型

`array`

.spec.mon.stretchCluster.zones[].volumeClaimTemplate.spec.selector.matchExpressions[]

描述

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

类型

`object`

必填

`key` `operator`

属性	类型	描述
`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.

属性	类型	描述
`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.mon.stretchCluster.zones[].volumeClaimTemplate.spec.selector.matchExpressions[].values

描述

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.

类型

`array`

.spec.mon.stretchCluster.zones[].volumeClaimTemplate.spec.selector.matchExpressions[].values[]

类型

`string`

.spec.mon.stretchCluster.zones[].volumeClaimTemplate.spec.selector.matchLabels

描述

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.

类型

`object`

.spec.mon.volumeClaimTemplate

描述

VolumeClaimTemplate is the PVC definition

类型

`object`

属性	类型	描述
`metadata`	`ObjectMeta`	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata ↗
`spec`	`object`	spec defines the desired characteristics of a volume requested by a pod author. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims ↗

.spec.mon.volumeClaimTemplate.spec

描述

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

类型

`object`

属性	类型	描述
`accessModes`	`array`	accessModes contains the desired access mod volume should have. More info: https://kubernetes.io/docs/concepts/storage/per volumes#access-modes-1 ↗
`dataSource`	`object`	dataSource field can be used to specify either: • An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) • An existing PVC (PersistentVolumeClaim) If provisioner or an external controller can sup the specified data source, it will create a nev volume based on the contents of the specifie source. When the AnyVolumeDataSource fe gate is enabled, dataSource contents will be to dataSourceRef, and dataSourceRef conte be copied to dataSource when dataSourceRef.namespace is not specified. namespace is specified, then dataSourceRe not be copied to dataSource.
`dataSourceRef`	`object`	dataSourceRef specifies the object from which populate the volume with data, if a non-empty v is desired. This may be any object from a non-e API group (non core object) or a PersistentVolumeClaim object. When this field i specified, volume binding will only succeed if th of the specified object matches some installed v populator or dynamic provisioner. This field will the functionality of the dataSource field and as both fields are non-empty, they must have the s

属性	类型	描述
		value. For backwards compatibility, when name isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to t same value automatically if one of them is empt the other is non-empty. When namespace is sp in dataSourceRef, dataSource isn't set to the sa value and must be empty. There are three impo differences between dataSource and dataSourc • While dataSource only allows two specific ty objects, dataSourceRef allows any non-core as well as PersistentVolumeClaim objects. • While dataSource ignores disallowed values (dropping them), dataSourceRef preserves a values, and generates an error if a disallowe value is specified. • While dataSource only allows local objects, dataSourceRef allows objects in any names (Beta) Using this field requires the AnyVolumeDataSource feature gate to be en (Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource featur to be enabled.
`resources`	`object`	resources represents the minimum resources th volume should have. Users are allowed to spec resource requirements that are lower than previ value but must still be higher than capacity recc the status field of the claim. More info: https://kubernetes.io/docs/concepts/storage/per volumes#resources ↗

属性	类型	描述
`selector`	`object`	selector is a label query over volumes to consid binding.
`storageClassName`	`string`	storageClassName is the name of the StorageC required by the claim. More info: https://kubernetes.io/docs/concepts/storage/per volumes#class-1 ↗
`volumeAttributesClassName`	`string`	volumeAttributesClassName may be used to se VolumeAttributesClass used by this claim. If spe the CSI driver will create or update the volume attributes defined in the corresponding VolumeAttributesClass. This has a different pur than storageClassName, it can be changed afte claim is created. An empty string or nil value inc that no VolumeAttributesClass will be applied to claim. If the claim enters an Infeasible error stat field can be reset to its previous value (including cancel the modification. If the resource referred volumeAttributesClass does not exist, this PersistentVolumeClaim will be set to a Pending as reflected by the modifyVolumeStatus field, u such as a resource exists. More info: https://kubernetes.io/docs/concepts/storage/vol attributes-classes/ ↗
`volumeMode`	`string`	volumeMode defines what type of volume is rec by the claim. Value of Filesystem is implied whe included in claim spec.

属性	类型	描述
`volumeName`	`string`	volumeName is the binding reference to the PersistentVolume backing this claim.

.spec.mon.volumeClaimTemplate.spec.accessModes

描述

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

类型

`array`

.spec.mon.volumeClaimTemplate.spec.accessModes[]

类型

`string`

.spec.mon.volumeClaimTemplate.spec.dataSource

描述

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

类型

`object`

必填

`kind` `name`

属性	类型	描述
`apiGroup`	`string`	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
`kind`	`string`	Kind is the type of resource being referenced
`name`	`string`	Name is the name of resource being referenced

.spec.mon.volumeClaimTemplate.spec.dataSourceRef

描述

dataSourceRef specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a PersistentVolumeClaim object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replace the functionality of the dataSource field and as such if both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in dataSourceRef, dataSource isn't set to the same value and must be empty. There are three important differences between dataSource and dataSourceRef: * While dataSource only allows two specific types of objects, dataSourceRef allows any non-core object, as well as PersistentVolumeClaim objects. * While dataSource ignores disallowed values (dropping them), dataSourceRef preserves all values, and generates an error if a disallowed value is specified. * While dataSource only allows local objects, dataSourceRef allows objects in any namespaces. (Beta) Using this field requires the AnyVolumeDataSource feature gate to be enabled.

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

类型

`object`

必填

`kind` `name`

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

.spec.mon.volumeClaimTemplate.spec.resources

描述

resources represents the minimum resources the volume should have. Users are allowed to specify resource requirements that are lower than previous value but must still be higher than capacity recorded in the status field of the claim. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources

类型

`object`

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

.spec.mon.volumeClaimTemplate.spec.resources.limits

描述

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

类型

`object`

.spec.mon.volumeClaimTemplate.spec.resources.requests

描述

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/

类型

`object`

.spec.mon.volumeClaimTemplate.spec.selector

描述

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

类型

`object`

属性	类型	描述
`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.mon.volumeClaimTemplate.spec.selector.matchExpressions

描述

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

类型

`array`

.spec.mon.volumeClaimTemplate.spec.selector.matchExpressions[]

描述

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

类型

`object`

必填

`key` `operator`

属性	类型	描述
`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.

属性	类型	描述
`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.mon.volumeClaimTemplate.spec.selector.matchExpressions[].values

描述

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.

类型

`array`

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

类型

`string`

.spec.mon.volumeClaimTemplate.spec.selector.matchLabels

描述

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.

类型

`object`

.spec.mon.zones

描述

Zones are specified when we want to provide zonal awareness to mons

类型

`array`

.spec.mon.zones[]

描述

MonZoneSpec represents the specification of a zone in a Ceph Cluster

类型

`object`

属性	类型	描述
`arbiter`	`boolean`	Arbiter determines if the zone contains the arbiter used for stretch cluster mode
`name`	`string`	Name is the name of the zone
`volumeClaimTemplate`	`object`	VolumeClaimTemplate is the PVC template

.spec.mon.zones[].volumeClaimTemplate

描述

VolumeClaimTemplate is the PVC template

类型

`object`

属性	类型	描述
`metadata`	`ObjectMeta`	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata ↗
`spec`	`object`	spec defines the desired characteristics of a volume requested by a pod author. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims ↗

.spec.mon.zones[].volumeClaimTemplate.spec

描述

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

类型

`object`

属性	类型	描述
`accessModes`	`array`	accessModes contains the desired access mod volume should have. More info: https://kubernetes.io/docs/concepts/storage/per volumes#access-modes-1 ↗
`dataSource`	`object`	dataSource field can be used to specify either:

属性	类型	描述
		• An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) • An existing PVC (PersistentVolumeClaim) If provisioner or an external controller can sup the specified data source, it will create a nev volume based on the contents of the specifie source. When the AnyVolumeDataSource fe gate is enabled, dataSource contents will be to dataSourceRef, and dataSourceRef conte be copied to dataSource when dataSourceRef.namespace is not specified. namespace is specified, then dataSourceRe not be copied to dataSource.
`dataSourceRef`	`object`	dataSourceRef specifies the object from which t populate the volume with data, if a non-empty v is desired. This may be any object from a non-e API group (non core object) or a PersistentVolumeClaim object. When this field i specified, volume binding will only succeed if th of the specified object matches some installed v populator or dynamic provisioner. This field will the functionality of the dataSource field and as s both fields are non-empty, they must have the s value. For backwards compatibility, when name isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to t same value automatically if one of them is empt the other is non-empty. When namespace is sp in dataSourceRef, dataSource isn't set to the sa value and must be empty. There are three impo differences between dataSource and dataSourc

属性	类型	描述
		• While dataSource only allows two specific ty objects, dataSourceRef allows any non-core as well as PersistentVolumeClaim objects. • While dataSource ignores disallowed values (dropping them), dataSourceRef preserves a values, and generates an error if a disallowe value is specified. • While dataSource only allows local objects, dataSourceRef allows objects in any names (Beta) Using this field requires the AnyVolumeDataSource feature gate to be en (Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource featur to be enabled.
`resources`	`object`	resources represents the minimum resources th volume should have. Users are allowed to spec resource requirements that are lower than previ value but must still be higher than capacity reco the status field of the claim. More info: https://kubernetes.io/docs/concepts/storage/per volumes#resources ↗
`selector`	`object`	selector is a label query over volumes to consid binding.

属性	类型	描述
`storageClassName`	`string`	storageClassName is the name of the StorageC required by the claim. More info: https://kubernetes.io/docs/concepts/storage/per volumes#class-1 ↗
`volumeAttributesClassName`	`string`	volumeAttributesClassName may be used to se VolumeAttributesClass used by this claim. If spe the CSI driver will create or update the volume v attributes defined in the corresponding VolumeAttributesClass. This has a different pur than storageClassName, it can be changed afte claim is created. An empty string or nil value inc that no VolumeAttributesClass will be applied to claim. If the claim enters an Infeasible error stat field can be reset to its previous value (including cancel the modification. If the resource referred volumeAttributesClass does not exist, this PersistentVolumeClaim will be set to a Pending as reflected by the modifyVolumeStatus field, u such as a resource exists. More info: https://kubernetes.io/docs/concepts/storage/vol attributes-classes/ ↗
`volumeMode`	`string`	volumeMode defines what type of volume is req by the claim. Value of Filesystem is implied whe included in claim spec.
`volumeName`	`string`	volumeName is the binding reference to the PersistentVolume backing this claim.

.spec.mon.zones[].volumeClaimTemplate.spec.accessModes

描述

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

类型

`array`

.spec.mon.zones[].volumeClaimTemplate.spec.accessModes[]

类型

`string`

.spec.mon.zones[].volumeClaimTemplate.spec.dataSource

描述

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

类型

`object`

必填

`kind` `name`

属性	类型	描述
`apiGroup`	`string`	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
`kind`	`string`	Kind is the type of resource being referenced
`name`	`string`	Name is the name of resource being referenced

.spec.mon.zones[].volumeClaimTemplate.spec.dataSourceRef

描述

dataSourceRef specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a PersistentVolumeClaim object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replace the functionality of the dataSource field and as such if both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in dataSourceRef, dataSource isn't set to the same value and must be empty. There are three important differences between dataSource and dataSourceRef: * While dataSource only allows two specific types of objects, dataSourceRef allows any non-core object, as well as PersistentVolumeClaim objects. * While dataSource ignores disallowed values (dropping them), dataSourceRef preserves all values, and generates an error if a disallowed value is specified. * While dataSource only allows local objects, dataSourceRef allows objects in any namespaces. (Beta) Using this field requires the AnyVolumeDataSource feature gate to be enabled.

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

类型

`object`

必填

`kind` `name`

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

.spec.mon.zones[].volumeClaimTemplate.spec.resources

描述

resources represents the minimum resources the volume should have. Users are allowed to specify resource requirements that are lower than previous value but must still be higher than capacity recorded in the status field of the claim. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources

类型

`object`

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

.spec.mon.zones[].volumeClaimTemplate.spec.resources.limits

描述

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

类型

`object`

.spec.mon.zones[].volumeClaimTemplate.spec.resources.requests

描述

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/

类型

`object`

.spec.mon.zones[].volumeClaimTemplate.spec.selector

描述

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

类型

`object`

属性	类型	描述
`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.mon.zones[].volumeClaimTemplate.spec.selector.matchExpressions

描述

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

类型

`array`

.spec.mon.zones[].volumeClaimTemplate.spec.selector.matchExpressions[]

描述

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

类型

`object`

必填

`key` `operator`

属性	类型	描述
`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.

属性	类型	描述
`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.mon.zones[].volumeClaimTemplate.spec.selector.matchExpressions[].values

描述

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.

类型

`array`

.spec.mon.zones[].volumeClaimTemplate.spec.selector.matchExpressions[].values[]

类型

`string`

.spec.mon.zones[].volumeClaimTemplate.spec.selector.matchLabels

描述

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.

类型

`object`

.spec.monitoring

描述

Prometheus based Monitoring settings

类型

`object`

属性	类型	描述
`enabled`	`boolean`	Enabled determines whether to create the prometheus rules for the ceph cluster. If true, the prometheus types must exist or the creation will fail. Default is false.
`exporter`	`object`	Ceph exporter configuration
`externalMgrEndpoints`	`array`	ExternalMgrEndpoints points to an existing Ceph prometheus exporter endpoint
`externalMgrPrometheusPort`	`integer`	ExternalMgrPrometheusPort Prometheus exporter port
`interval`	`string`	Interval determines prometheus scrape interval

属性	类型	描述
`metricsDisabled`	`boolean`	Whether to disable the metrics reported by Ceph. If false, the prometheus mgr module and Ceph exporter are enabled. If true, the prometheus mgr module and Ceph exporter are both disabled. Default is false.
`port`	`integer`	Port is the prometheus server port

.spec.monitoring.exporter

描述

Ceph exporter configuration

类型

`object`

属性	类型	描述
`hostNetwork`	`boolean`	Whether host networking is enabled for CephExporter. If not set, the network settings from CephCluster.spec.networking will be applied.
`perfCountersPrioLimit`	`integer`	Only performance counters greater than or equal to this option are fetched

属性	类型	描述
`statsPeriodSeconds`	`integer`	Time to wait before sending requests again to exporter server (seconds)

.spec.monitoring.externalMgrEndpoints

描述

ExternalMgrEndpoints points to an existing Ceph prometheus exporter endpoint

类型

`array`

.spec.monitoring.externalMgrEndpoints[]

描述

EndpointAddress is a tuple that describes single IP address. Deprecated: This API is deprecated in v1.33+.

类型

`object`

必填

`ip`

属性	类型	描述
`hostname`	`string`	The Hostname of this endpoint
`ip`	`string`	The IP of this endpoint. May not be loopback (127.0.0.0/8 or ::1), link-local (169.254.0.0/16 or fe80::/10), or link-local multicast (224.0.0.0/24 or ff02::/16).

属性	类型	描述
`nodeName`	`string`	Optional: Node hosting this endpoint. This can be used to determine endpoints local to a node.
`targetRef`	`object`	Reference to object providing the endpoint.

.spec.monitoring.externalMgrEndpoints[].targetRef

描述

Reference to object providing the endpoint.

类型

`object`

属性	类型	描述
`apiVersion`	`string`	API version of the referent.
`fieldPath`	`string`	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 desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is 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.

属性	类型	描述
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.spec.network

描述

Network related configuration

类型

`object`

属性	类型	描述
`addressRanges`	`object`	AddressRanges specify a list of CIDRs that Rook will a to Ceph's 'public_network' and/or 'cluster_network' configurations. This config section may be used for the "host" or "multus" network providers.
`connections`	`object`	Settings for network connections such as compression encryption across the wire.
`dualStack`	`boolean`	DualStack determines whether Ceph daemons should on both IPv4 and IPv6
`hostNetwork`	`boolean`	HostNetwork to enable host network. If host networkin enabled or disabled on a running cluster, then the oper will automatically fail over all the mons to apply the nev network settings.
`ipFamily`	`string`	IPFamily is the single stack IPv6 or IPv4 protocol
`multiClusterService`	`object`	Enable multiClusterService to export the Services betw peer clusters
`provider`	`string`	Provider is what provides network connectivity to the c e.g. "host" or "multus". If the Provider is updated from I

属性	类型	描述
		empty to "host" on a running cluster, then the operator automatically fail over all the mons to apply the "host" network settings.
`selectors`	`object`	Selectors define NetworkAttachmentDefinitions to be u for Ceph public and/or cluster networks when the "mult network provider is used. This config section is not use other network providers. Valid keys are "public" and "cluster". Refer to Ceph networking documentation for more: https://docs.ceph.com/en/latest/rados/configuration/ne config-ref/ ↗ Refer to Multus network annotation documentation for selecting values: https://github.com/k8snetworkplumbingwg/multus-cni/blob/master/docs/how-to-use.md#run-pod-with-net annotation ↗ Rook will make a best-effort attempt to automatically d CIDR address ranges for given network attachment definitions. Rook's methods are robust but may be imprecise for sufficiently complicated networks. Rook's auto-detection process obtains a new IP address lease each CephCluster reconcile. If Rook fails to detect, incorrectly detects, only partially detects, or if underlyir networks do not support reusing old IP addresses, it is to use the 'addressRanges' config section to specify C ranges for the Ceph cluster. As a contrived example, one can use a theoretical Kubernetes-wide network for Ceph client traffic and a theoretical Rook-only network for Ceph replication traff

属性	类型	描述
		shown: selectors: public: "default/cluster-fast-net" clust "rook-ceph/ceph-backend-net"

.spec.network.addressRanges

描述

AddressRanges specify a list of CIDRs that Rook will apply to Ceph's 'public_network' and/or 'cluster_network' configurations. This config section may be used for the "host" or "multus" network providers.

类型

`object`

属性	类型	描述
`cluster`	`array`	Cluster defines a list of CIDRs to use for Ceph cluster network communication.
`public`	`array`	Public defines a list of CIDRs to use for Ceph public network communication.

.spec.network.addressRanges.cluster

描述

Cluster defines a list of CIDRs to use for Ceph cluster network communication.

类型

`array`

.spec.network.addressRanges.cluster[]

描述

An IPv4 or IPv6 network CIDR. This naive kubebuilder regex provides immediate feedback for some typos and for a common problem case where the range spec is forgotten (e.g., /24). Rook does in-depth validation in code.

类型

`string`

.spec.network.addressRanges.public

描述

Public defines a list of CIDRs to use for Ceph public network communication.

类型

`array`

.spec.network.addressRanges.public[]

描述

An IPv4 or IPv6 network CIDR. This naive kubebuilder regex provides immediate feedback for some typos and for a common problem case where the range spec is forgotten (e.g., /24). Rook does in-depth validation in code.

类型

`string`

.spec.network.connections

描述

Settings for network connections such as compression and encryption across the wire.

类型

`object`

属性	类型	描述
`compression`	`object`	Compression settings for the network connections.
`encryption`	`object`	Encryption settings for the network connections.
`requireMsgr2`	`boolean`	Whether to require msgr2 (port 3300) even if compression or encryption are not enabled. If true, the msgr1 port (6789) will be disabled. Requires a kernel that supports msgr2 (kernel 5.11 or CentOS 8.4 or newer).

.spec.network.connections.compression

描述

Compression settings for the network connections.

类型

`object`

属性	类型	描述
`enabled`	`boolean`	Whether to compress the data in transit across the wire. The default is not set.

.spec.network.connections.encryption

描述

Encryption settings for the network connections.

类型

`object`

属性	类型	描述
`enabled`	`boolean`	Whether to encrypt the data in transit across the wire to prevent eavesdropping the data on the network. The default is not set. Even if encryption is not enabled, clients still establish a strong initial authentication for the connection and data integrity is still validated with a crc check. When encryption is enabled, all communication between clients and Ceph daemons, or between Ceph daemons will be encrypted.

.spec.network.multiClusterService

描述

Enable multiClusterService to export the Services between peer clusters

类型

`object`

属性	类型	描述
`clusterID`	`string`	ClusterID uniquely identifies a cluster. It is used as a prefix to nslookup exported services. For example: ...svc.clusterset.local
`enabled`	`boolean`	Enable multiClusterService to export the mon and OSD services to peer cluster. Ensure that peer clusters are connected using an MCS API compatible application, like Globalnet Submariner.

.spec.network.selectors

描述

Selectors define NetworkAttachmentDefinitions to be used for Ceph public and/or cluster networks when the "multus" network provider is used. This config section is not used for other network providers. Valid keys are "public" and "cluster". Refer to Ceph networking documentation for more: https://docs.ceph.com/en/latest/rados/configuration/network-config-ref/ Refer to Multus network annotation documentation for help selecting values: https://github.com/k8snetworkplumbingwg/multus-cni/blob/master/docs/how-to-use.md#run-pod-with-network-annotation Rook will make a best-effort attempt to automatically detect CIDR address ranges for given network attachment definitions. Rook's methods are robust but may be imprecise for sufficiently complicated networks. Rook's auto-detection process obtains a new IP address lease for each CephCluster reconcile. If Rook fails to detect, incorrectly detects, only partially detects, or if underlying networks do not support reusing old IP addresses, it is best to use the 'addressRanges' config section to specify CIDR ranges for the Ceph cluster. As a contrived example, one can use a theoretical Kubernetes-wide network for Ceph client traffic and a theoretical Rook-only network for Ceph replication traffic as shown: selectors: public: "default/cluster-fast-net" cluster: "rook-ceph/ceph-backend-net"

类型

`object`

.spec.placement

类型

`object`

.spec.priorityClassNames

描述

PriorityClassNames sets priority classes on components

类型

`object`

.spec.resources

描述

Resources set resource requests and limits

类型

`object`

.spec.security

描述

Security represents security settings

类型

`object`

属性	类型	描述
`cephx`	`object`	CephX configures CephX key settings. More: https://docs.ceph.com/en/latest/dev/cephx/ ↗
`keyRotation`	`object`	KeyRotation defines options for rotation of OSD disk encryption keys.
`kms`	`object`	KeyManagementService is the main Key Management option

.spec.security.cephx

描述

CephX configures CephX key settings. More: https://docs.ceph.com/en/latest/dev/cephx/

类型

`object`

属性	类型	描述
`csi`	`object`	CSI configures CephX key rotation settings for the Ceph-CSI daemons in the current Kubernetes cluster. CSI key rotation can affect existing PV connections, so take care when exercising this option.
`daemon`	`object`	Daemon configures CephX key settings for local Ceph daemons managed by Rook and part of the Ceph cluster. Daemon CephX keys can be rotated without affecting client connections.
`rbdMirrorPeer`	`object`	RBDMirrorPeer configures CephX key settings of the `rbd-mirror-peer` user that is used for creating bootstrap peer token used connect peer clusters. Rotating the `rbd-mirror-peer` user key will update the mirror peer token. Rotation will affect any existing peers connected to this cluster, so take care when exercising this option.

.spec.security.cephx.csi

描述

CSI configures CephX key rotation settings for the Ceph-CSI daemons in the current Kubernetes cluster. CSI key rotation can affect existing PV connections, so take care when exercising this option.

类型

`object`

属性	类型	描述
`keepPriorKeyCountMax`	`integer`	KeepPriorKeyCountMax tells Rook how many prior keys to keep active. Generally, this would be set to 1 to allow for a migration period for applications. If desired, set this to 0 to delete prior keys after migration. This config only applies to prior keys that already exist. If PriorKeyCount is set to 2 while only a single key currently exists, only a single prior key will be kept, and the reported status will only indicate the actual number of prior keys, not necessarily a reflection of PriorKeyCount config here.
`keyGeneration`	`integer`	KeyGeneration specifies the desired CephX key generation. This is used when KeyRotationPolicy is KeyGeneration and ignored for other policies. If this is set to greater than the current key generation, relevant keys will be rotated, and the generation value will be updated to this new value (generation values are not necessarily incremental, though that is the intended use case). If this is set to less than or equal to the current key generation, keys are not rotated.
`keyRotationPolicy`	`string`	KeyRotationPolicy controls if and when CephX keys are rotated after initial creation. One of Disabled, or KeyGeneration. Default Disabled.

.spec.security.cephx.daemon

描述

Daemon configures CephX key settings for local Ceph daemons managed by Rook and part of the Ceph cluster. Daemon CephX keys can be rotated without affecting client connections.

类型

`object`

属性	类型	描述
`keyGeneration`	`integer`	KeyGeneration specifies the desired CephX key generation. This is used when KeyRotationPolicy is KeyGeneration and ignored for other policies. If this is set to greater than the current key generation, relevant keys will be rotated, and the generation value will be updated to this new value (generation values are not necessarily incremental, though that is the intended use case). If this is set to less than or equal to the current key generation, keys are not rotated.
`keyRotationPolicy`	`string`	KeyRotationPolicy controls if and when CephX keys are rotated after initial creation. One of Disabled, or KeyGeneration. Default Disabled.

.spec.security.cephx.rbdMirrorPeer

描述

RBDMirrorPeer configures CephX key settings of the `rbd-mirror-peer` user that is used for creating bootstrap peer token used connect peer clusters. Rotating the `rbd-mirror-peer` user key will update the mirror peer token. Rotation will affect any existing peers connected to this cluster, so take care when exercising this option.

类型

`object`

属性	类型	描述
`keyGeneration`	`integer`	KeyGeneration specifies the desired CephX key generation. This is used when KeyRotationPolicy is KeyGeneration and ignored for other policies. If this is set to greater than the current key generation, relevant keys will be rotated, and the generation value will be updated to this new value (generation values are not necessarily incremental, though that is the intended use case). If this is set to less than or equal to the current key generation, keys are not rotated.
`keyRotationPolicy`	`string`	KeyRotationPolicy controls if and when CephX keys are rotated after initial creation. One of Disabled, or KeyGeneration. Default Disabled.

.spec.security.keyRotation

描述

KeyRotation defines options for rotation of OSD disk encryption keys.

类型

`object`

属性	类型	描述
`enabled`	`boolean`	Enabled represents whether the key rotation is enabled.

属性	类型	描述
`schedule`	`string`	Schedule represents the cron schedule for key rotation.

.spec.security.kms

描述

KeyManagementService is the main Key Management option

类型

`object`

属性	类型	描述
`connectionDetails`	`object`	ConnectionDetails contains the KMS connection details (address, port etc)
`tokenSecretName`	`string`	TokenSecretName is the kubernetes secret containing the KMS token

.spec.security.kms.connectionDetails

描述

ConnectionDetails contains the KMS connection details (address, port etc)

类型

`object`

.spec.storage

描述

A spec for available storage in the cluster and how it should be used

类型

`object`

属性	类型	描述
`allowDeviceClassUpdate`	`boolean`	Whether to allow updating the device class after the OSD is initially provisioned
`allowOsdCrushWeightUpdate`	`boolean`	Whether Rook will resize the OSD CRUSH weight when the OSD PVC size is increased. This allows cluster data to be rebalanced to make most effective use of new OSD space. The default is false since data rebalancing can cause temporary cluster slowdown.
`backfillFullRatio`	`number`	BackfillFullRatio is the ratio at which the cluster is too full for backfill. Backfill will be disabled if above this threshold. Default is 0.90.
`config`	`object`	
`deviceFilter`	`string`	A regular expression to allow more fine-grained selection of devices on nodes across the cluster

属性	类型	描述
`devicePathFilter`	`string`	A regular expression to allow more fine-grained selection of devices with path names
`devices`	`array`	List of devices to use as storage devices
`flappingRestartIntervalHours`	`integer`	FlappingRestartIntervalHours defines the time for which the OSD pods, that failed with zero exit code, will sleep before restarting. This is needed for OSD flapping where OSD daemons are marked down more than 5 times in 600 seconds by Ceph. Preventing the OSD pods to restart immediately in such scenarios will prevent Rook from marking OSD as `up` and thus peering of the PGs mapped to the OSD. User needs to manually restart the OSD pod if they manage to fix the underlying OSD flapping issue before the restart interval. The sleep will be disabled if this interval is set to 0.
`fullRatio`	`number`	FullRatio is the ratio at which the cluster is considered full and ceph will stop accepting writes. Default is 0.95.

属性	类型	描述
`migration`	`object`	Migration handles the OSD migration
`nearFullRatio`	`number`	NearFullRatio is the ratio at which the cluster is considered nearly full and will raise a ceph health warning. Default is 0.85.
`nodes`	`array`	
`onlyApplyOSDPlacement`	`boolean`	
`osdMaxUpdatesInParallel`	`integer`	The maximum number of OSDs to update in parallel.
`scheduleAlways`	`boolean`	Whether to always schedule OSDs on a node even if the node is not currently scheduleable or ready
`storageClassDeviceSets`	`array`	
`store`	`object`	OSDStore is the backend storage type used for creating the OSDs
`useAllDevices`	`boolean`	Whether to consume all the storage devices found on a machine
`useAllNodes`	`boolean`	

属性	类型	描述
`volumeClaimTemplates`	`array`	PersistentVolumeClaims to use as storage

.spec.storage.config

类型

`object`

.spec.storage.devices

描述

List of devices to use as storage devices

类型

`array`

.spec.storage.devices[]

描述

Device represents a disk to use in the cluster

类型

`object`

属性	类型	描述
`config`	`object`	
`fullpath`	`string`	
`name`	`string`	

.spec.storage.devices[].config

类型

`object`

.spec.storage.migration

描述

Migration handles the OSD migration

类型

`object`

属性	类型	描述
`confirmation`	`string`	A user confirmation to migrate the OSDs. It destroys each OSD one at a time, cleans up the backing disk and prepares OSD with same ID on that disk

.spec.storage.nodes

类型

`array`

.spec.storage.nodes[]

描述

Node is a storage nodes

类型

`object`

属性	类型	描述
`config`	`object`	
`deviceFilter`	`string`	A regular expression to allow more fine-grained selection of devices on nodes across the cluster
`devicePathFilter`	`string`	A regular expression to allow more fine-grained selection of devices with path names
`devices`	`array`	List of devices to use as storage devices
`name`	`string`	
`resources`	`object`	ResourceRequirements describes the compute resource requirements.
`useAllDevices`	`boolean`	Whether to consume all the storage devices found on a machine
`volumeClaimTemplates`	`array`	PersistentVolumeClaims to use as storage

.spec.storage.nodes[].config

类型

`object`

.spec.storage.nodes[].devices

描述

List of devices to use as storage devices

类型

`array`

.spec.storage.nodes[].devices[]

描述

Device represents a disk to use in the cluster

类型

`object`

属性	类型	描述
`config`	`object`	
`fullpath`	`string`	
`name`	`string`	

.spec.storage.nodes[].devices[].config

类型

`object`

.spec.storage.nodes[].resources

描述

ResourceRequirements describes the compute resource requirements.

类型

`object`

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

.spec.storage.nodes[].resources.claims

描述

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

类型

`array`

.spec.storage.nodes[].resources.claims[]

描述

ResourceClaim references one entry in PodSpec.ResourceClaims.

类型

`object`

必填

`name`

属性	类型	描述
`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.storage.nodes[].resources.limits

描述

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

类型

`object`

.spec.storage.nodes[].resources.requests

描述

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/

类型

`object`

.spec.storage.nodes[].volumeClaimTemplates

描述

PersistentVolumeClaims to use as storage

类型

`array`

.spec.storage.nodes[].volumeClaimTemplates[]

描述

VolumeClaimTemplate is a simplified version of K8s corev1's PVC. It has no type meta or status.

类型

`object`

属性	类型	描述
`metadata`	`ObjectMeta`	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata ↗

属性	类型	描述
`spec`	`object`	spec defines the desired characteristics of a volume requested by a pod author. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims ↗

.spec.storage.nodes[].volumeClaimTemplates[].spec

描述

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

类型

`object`

属性	类型	描述
`accessModes`	`array`	accessModes contains the desired access mod volume should have. More info: https://kubernetes.io/docs/concepts/storage/per volumes#access-modes-1 ↗
`dataSource`	`object`	dataSource field can be used to specify either: • An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) • An existing PVC (PersistentVolumeClaim) If provisioner or an external controller can sup the specified data source, it will create a nev volume based on the contents of the specifie source. When the AnyVolumeDataSource fe gate is enabled, dataSource contents will be

属性	类型	描述
		to dataSourceRef, and dataSourceRef conte be copied to dataSource when dataSourceRef.namespace is not specified. namespace is specified, then dataSourceRe not be copied to dataSource.
`dataSourceRef`	`object`	dataSourceRef specifies the object from which t populate the volume with data, if a non-empty v is desired. This may be any object from a non-e API group (non core object) or a PersistentVolumeClaim object. When this field i specified, volume binding will only succeed if th of the specified object matches some installed v populator or dynamic provisioner. This field will the functionality of the dataSource field and as s both fields are non-empty, they must have the s value. For backwards compatibility, when name isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to t same value automatically if one of them is empt the other is non-empty. When namespace is sp in dataSourceRef, dataSource isn't set to the sa value and must be empty. There are three impo differences between dataSource and dataSourc • While dataSource only allows two specific ty objects, dataSourceRef allows any non-core as well as PersistentVolumeClaim objects. • While dataSource ignores disallowed values (dropping them), dataSourceRef preserves a values, and generates an error if a disallowe value is specified. • While dataSource only allows local objects, dataSourceRef allows objects in any names

属性	类型	描述
		(Beta) Using this field requires the AnyVolumeDataSource feature gate to be en (Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource featur to be enabled.
`resources`	`object`	resources represents the minimum resources th volume should have. Users are allowed to spec resource requirements that are lower than previ value but must still be higher than capacity recc the status field of the claim. More info: https://kubernetes.io/docs/concepts/storage/per volumes#resources ↗
`selector`	`object`	selector is a label query over volumes to consid binding.
`storageClassName`	`string`	storageClassName is the name of the StorageC required by the claim. More info: https://kubernetes.io/docs/concepts/storage/per volumes#class-1 ↗
`volumeAttributesClassName`	`string`	volumeAttributesClassName may be used to se VolumeAttributesClass used by this claim. If spe the CSI driver will create or update the volume v attributes defined in the corresponding VolumeAttributesClass. This has a different pur than storageClassName, it can be changed afte claim is created. An empty string or nil value inc

属性	类型	描述
		that no VolumeAttributesClass will be applied to claim. If the claim enters an Infeasible error stat field can be reset to its previous value (including cancel the modification. If the resource referred volumeAttributesClass does not exist, this PersistentVolumeClaim will be set to a Pending as reflected by the modifyVolumeStatus field, u such as a resource exists. More info: https://kubernetes.io/docs/concepts/storage/vol attributes-classes/ ↗
`volumeMode`	`string`	volumeMode defines what type of volume is rec by the claim. Value of Filesystem is implied whe included in claim spec.
`volumeName`	`string`	volumeName is the binding reference to the PersistentVolume backing this claim.

.spec.storage.nodes[].volumeClaimTemplates[].spec.accessModes

描述

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

类型

`array`

.spec.storage.nodes[].volumeClaimTemplates[].spec.accessModes[]

类型

`string`

.spec.storage.nodes[].volumeClaimTemplates[].spec.dataSource

描述

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

类型

`object`

必填

`kind` `name`

属性	类型	描述
`apiGroup`	`string`	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
`kind`	`string`	Kind is the type of resource being referenced

属性	类型	描述
`name`	`string`	Name is the name of resource being referenced

.spec.storage.nodes[].volumeClaimTemplates[].spec.dataSourceRef

描述

dataSourceRef specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a PersistentVolumeClaim object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replace the functionality of the dataSource field and as such if both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in dataSourceRef, dataSource isn't set to the same value and must be empty. There are three important differences between dataSource and dataSourceRef: * While dataSource only allows two specific types of objects, dataSourceRef allows any non-core object, as well as PersistentVolumeClaim objects. * While dataSource ignores disallowed values (dropping them), dataSourceRef preserves all values, and generates an error if a disallowed value is specified. * While dataSource only allows local objects, dataSourceRef allows objects in any namespaces. (Beta) Using this field requires the AnyVolumeDataSource feature gate to be enabled. (Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

类型

`object`

必填

`kind` `name`

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

.spec.storage.nodes[].volumeClaimTemplates[].spec.resources

描述

resources represents the minimum resources the volume should have. Users are allowed to specify resource requirements that are lower than previous value but must still be higher than capacity recorded in the status field of the claim. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources

类型

`object`

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

.spec.storage.nodes[].volumeClaimTemplates[].spec.resources.limits

描述

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

类型

`object`

.spec.storage.nodes[].volumeClaimTemplates[].spec.resources.requests

描述

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/

类型

`object`

.spec.storage.nodes[].volumeClaimTemplates[].spec.selector

描述

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

类型

`object`

属性	类型	描述
`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.storage.nodes[].volumeClaimTemplates[].spec.selector.matchExpressions

描述

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

类型

`array`

.spec.storage.nodes[].volumeClaimTemplates[].spec.selector.matchExpressions[]

描述

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

类型

`object`

必填

`key` `operator`

属性	类型	描述
`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.storage.nodes[].volumeClaimTemplates[].spec.selector.matchExpressions[].values

描述

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.

类型

`array`

.spec.storage.nodes[].volumeClaimTemplates[].spec.selector.matchExpressions[].values[]

类型

`string`

.spec.storage.nodes[].volumeClaimTemplates[].spec.selector.matchLabels

描述

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.

类型

`object`

.spec.storage.storageClassDeviceSets

类型

`array`

.spec.storage.storageClassDeviceSets[]

描述

StorageClassDeviceSet is a storage class device set

类型

`object`

必填

`count` `name` `volumeClaimTemplates`

属性	类型	描述
`config`	`object`	Provider-specific device configuration
`count`	`integer`	Count is the number of devices in this set
`encrypted`	`boolean`	Whether to encrypt the deviceSet
`name`	`string`	Name is a unique identifier for the set
`placement`	`object`	
`portable`	`boolean`	Portable represents OSD portability across the hosts
`preparePlacement`	`object`	
`resources`	`object`	ResourceRequirements describes the compute resource requirements.
`schedulerName`	`string`	Scheduler name for OSD pod placement

属性	类型	描述
`tuneDeviceClass`	`boolean`	TuneSlowDeviceClass Tune the OSD when running on a slow Device Class
`tuneFastDeviceClass`	`boolean`	TuneFastDeviceClass Tune the OSD when running on a fast Device Class
`volumeClaimTemplates`	`array`	VolumeClaimTemplates is a list of PVC templates for the underlying storage devices

.spec.storage.storageClassDeviceSets[].config

描述

Provider-specific device configuration

类型

`object`

.spec.storage.storageClassDeviceSets[].placement

类型

`object`

属性	类型	描述
`nodeAffinity`	`object`	
`podAffinity`	`object`	
`podAntiAffinity`	`object`	

属性	类型	描述
`tolerations`	`array`	
`topologySpreadConstraints`	`array`	

.spec.storage.storageClassDeviceSets[].placement.nodeAffinity

类型

`object`

属性	类型	描述
`preferredDuringSchedulingIgnoredDuringExecution`	`array`	
`requiredDuringSchedulingIgnoredDuringExecution`	`object`	

.spec.storage.storageClassDeviceSets[].placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

类型

`object`

必填

`preference` `weight`

属性	类型	描述
`preference`	`object`	
`weight`	`integer`	

.spec.storage.storageClassDeviceSets[].placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchFields`	`array`	

.spec.storage.storageClassDeviceSets[].placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.storage.storageClassDeviceSets[].placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values[]

类型

`string`

.spec.storage.storageClassDeviceSets[].placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.storage.storageClassDeviceSets[].placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values[]

类型

`string`

.spec.storage.storageClassDeviceSets[].placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution

类型

`object`

必填

`nodeSelectorTerms`

属性	类型	描述
`nodeSelectorTerms`	`array`	

.spec.storage.storageClassDeviceSets[].placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[]

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchFields`	`array`	

.spec.storage.storageClassDeviceSets[].placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.storage.storageClassDeviceSets[].placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values[]

类型

`string`

.spec.storage.storageClassDeviceSets[].placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.storage.storageClassDeviceSets[].placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values[]

类型

`string`

.spec.storage.storageClassDeviceSets[].placement.podAffinity

类型

`object`

属性	类型	描述
`preferredDuringSchedulingIgnoredDuringExecution`	`array`	
`requiredDuringSchedulingIgnoredDuringExecution`	`array`	

.spec.storage.storageClassDeviceSets[].placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

类型

`object`

必填

`podAffinityTerm` `weight`

属性	类型	描述
`podAffinityTerm`	`object`	
`weight`	`integer`	

.spec.storage.storageClassDeviceSets[].placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm

类型

`object`

必填

`topologyKey`

属性	类型	描述
`labelSelector`	`object`	
`matchLabelKeys`	`array`	
`mismatchLabelKeys`	`array`	
`namespaceSelector`	`object`	
`namespaces`	`array`	

属性	类型	描述
`topologyKey`	`string`	

.spec.storage.storageClassDeviceSets[].placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.storage.storageClassDeviceSets[].placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.storage.storageClassDeviceSets[].placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]

类型

`string`

.spec.storage.storageClassDeviceSets[].placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels

类型

`object`

.spec.storage.storageClassDeviceSets[].placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]

类型

`string`

.spec.storage.storageClassDeviceSets[].placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]

类型

`string`

.spec.storage.storageClassDeviceSets[].placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.storage.storageClassDeviceSets[].placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.storage.storageClassDeviceSets[].placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]

类型

`string`

.spec.storage.storageClassDeviceSets[].placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

类型

`object`

.spec.storage.storageClassDeviceSets[].placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

类型

`string`

.spec.storage.storageClassDeviceSets[].placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

类型

`object`

必填

`topologyKey`

属性	类型	描述
`labelSelector`	`object`	

属性	类型	描述
`matchLabelKeys`	`array`	
`mismatchLabelKeys`	`array`	
`namespaceSelector`	`object`	
`namespaces`	`array`	
`topologyKey`	`string`	

.spec.storage.storageClassDeviceSets[].placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.storage.storageClassDeviceSets[].placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.storage.storageClassDeviceSets[].placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]

类型

`string`

.spec.storage.storageClassDeviceSets[].placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels

类型

`object`

.spec.storage.storageClassDeviceSets[].placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]

类型

`string`

.spec.storage.storageClassDeviceSets[].placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]

类型

`string`

.spec.storage.storageClassDeviceSets[].placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.storage.storageClassDeviceSets[].placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.storage.storageClassDeviceSets[].placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]

类型

`string`

.spec.storage.storageClassDeviceSets[].placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels

类型

`object`

.spec.storage.storageClassDeviceSets[].placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]

类型

`string`

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity

类型

`object`

属性	类型	描述
`preferredDuringSchedulingIgnoredDuringExecution`	`array`	
`requiredDuringSchedulingIgnoredDuringExecution`	`array`	

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

类型

`object`

必填

`podAffinityTerm` `weight`

属性	类型	描述
`podAffinityTerm`	`object`	
`weight`	`integer`	

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm

类型

`object`

必填

`topologyKey`

属性	类型	描述
`labelSelector`	`object`	
`matchLabelKeys`	`array`	

属性	类型	描述
`mismatchLabelKeys`	`array`	
`namespaceSelector`	`object`	
`namespaces`	`array`	
`topologyKey`	`string`	

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecuti

on[].podAffinityTerm.labelSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]

类型

`string`

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels

类型

`object`

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]

类型

`string`

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]

类型

`string`

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecuti

on[].podAffinityTerm.namespaceSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]

类型

`string`

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

类型

`object`

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

类型

`string`

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

类型

`object`

必填

`topologyKey`

属性	类型	描述
`labelSelector`	`object`	
`matchLabelKeys`	`array`	
`mismatchLabelKeys`	`array`	
`namespaceSelector`	`object`	
`namespaces`	`array`	
`topologyKey`	`string`	

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	

属性	类型	描述
`matchLabels`	`object`	

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]

类型

`string`

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels

类型

`object`

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]

类型

`string`

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]

类型

`string`

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]

类型

`string`

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels

类型

`object`

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]

类型

`string`

.spec.storage.storageClassDeviceSets[].placement.tolerations

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.tolerations[]

类型

`object`

属性	类型	描述
`effect`	`string`	
`key`	`string`	
`operator`	`string`	
`tolerationSeconds`	`integer`	
`value`	`string`	

.spec.storage.storageClassDeviceSets[].placement.topologySpreadConstraints

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.topologySpreadConstraints[]

类型

`object`

必填

`maxSkew` `topologyKey` `whenUnsatisfiable`

属性	类型	描述
`labelSelector`	`object`	
`matchLabelKeys`	`array`	
`maxSkew`	`integer`	
`minDomains`	`integer`	
`nodeAffinityPolicy`	`string`	
`nodeTaintsPolicy`	`string`	
`topologyKey`	`string`	
`whenUnsatisfiable`	`string`	

.spec.storage.storageClassDeviceSets[].placement.topologySpreadConstraints[].labelSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.storage.storageClassDeviceSets[].placement.topologySpreadConstraints[].labelSelector.matchExpressions

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.topologySpreadConstraints[].labelSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.storage.storageClassDeviceSets[].placement.topologySpreadConstraints[].labelSelector.matchExpressions[].values

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.topologySpreadConstraints[].labelSelector.matchExpressions[].values[]

类型

`string`

.spec.storage.storageClassDeviceSets[].placement.topologySpreadConstraints[].labelSelector.matchLabels

类型

`object`

.spec.storage.storageClassDeviceSets[].placement.topologySpreadConstraints[].matchLabelKeys

类型

`array`

.spec.storage.storageClassDeviceSets[].placement.topologySpreadConstraints[].matchLabelKeys[]

类型

`string`

.spec.storage.storageClassDeviceSets[].preparePlacement

类型

`object`

属性	类型	描述
`nodeAffinity`	`object`	
`podAffinity`	`object`	
`podAntiAffinity`	`object`	
`tolerations`	`array`	

属性	类型	描述
`topologySpreadConstraints`	`array`	

.spec.storage.storageClassDeviceSets[].preparePlacement.nodeAffinity

类型

`object`

属性	类型	描述
`preferredDuringSchedulingIgnoredDuringExecution`	`array`	
`requiredDuringSchedulingIgnoredDuringExecution`	`object`	

.spec.storage.storageClassDeviceSets[].preparePlacement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

类型

`object`

必填

`preference` `weight`

属性	类型	描述
`preference`	`object`	
`weight`	`integer`	

.spec.storage.storageClassDeviceSets[].preparePlacement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchFields`	`array`	

.spec.storage.storageClassDeviceSets[].preparePlacement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.storage.storageClassDeviceSets[].preparePlacement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values[]

类型

`string`

.spec.storage.storageClassDeviceSets[].preparePlacement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.storage.storageClassDeviceSets[].preparePlacement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values[]

类型

`string`

.spec.storage.storageClassDeviceSets[].preparePlacement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution

类型

`object`

必填

`nodeSelectorTerms`

属性	类型	描述
`nodeSelectorTerms`	`array`	

.spec.storage.storageClassDeviceSets[].preparePlacement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[]

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	

属性	类型	描述
`matchFields`	`array`	

.spec.storage.storageClassDeviceSets[].preparePlacement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.storage.storageClassDeviceSets[].preparePlacement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values[]

类型

`string`

.spec.storage.storageClassDeviceSets[].preparePlacement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	

属性	类型	描述
`values`	`array`	

.spec.storage.storageClassDeviceSets[].preparePlacement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values[]

类型

`string`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity

类型

`object`

属性	类型	描述
`preferredDuringSchedulingIgnoredDuringExecution`	`array`	
`requiredDuringSchedulingIgnoredDuringExecution`	`array`	

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

类型

`object`

必填

`podAffinityTerm` `weight`

属性	类型	描述
`podAffinityTerm`	`object`	
`weight`	`integer`	

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm

类型

`object`

必填

`topologyKey`

属性	类型	描述
`labelSelector`	`object`	
`matchLabelKeys`	`array`	
`mismatchLabelKeys`	`array`	
`namespaceSelector`	`object`	
`namespaces`	`array`	
`topologyKey`	`string`	

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]

类型

`string`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels

类型

`object`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]

类型

`string`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]

类型

`string`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.preferredDuringSchedulingIgnoredDuringExe

cution[].podAffinityTerm.namespaceSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]

类型

`string`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

类型

`object`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

类型

`string`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

类型

`object`

必填

`topologyKey`

属性	类型	描述
`labelSelector`	`object`	
`matchLabelKeys`	`array`	
`mismatchLabelKeys`	`array`	
`namespaceSelector`	`object`	
`namespaces`	`array`	
`topologyKey`	`string`	

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	

属性	类型	描述
`matchLabels`	`object`	

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]

类型

`string`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels

类型

`object`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]

类型

`string`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]

类型

`string`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]

类型

`string`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels

类型

`object`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]

类型

`string`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity

类型

`object`

属性	类型	描述
`preferredDuringSchedulingIgnoredDuringExecution`	`array`	
`requiredDuringSchedulingIgnoredDuringExecution`	`array`	

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

类型

`object`

必填

`podAffinityTerm` `weight`

属性	类型	描述
`podAffinityTerm`	`object`	
`weight`	`integer`	

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm

类型

`object`

必填

`topologyKey`

属性	类型	描述
`labelSelector`	`object`	
`matchLabelKeys`	`array`	
`mismatchLabelKeys`	`array`	
`namespaceSelector`	`object`	
`namespaces`	`array`	
`topologyKey`	`string`	

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.preferredDuringSchedulingIgnoredDurin

gExecution[].podAffinityTerm.labelSelector.matchExpressions

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]

类型

`string`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels

类型

`object`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]

类型

`string`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]

类型

`string`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.preferredDuringSchedulingIgnoredDuring

gExecution[].podAffinityTerm.namespaceSelector.matchExpressions

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]

类型

`string`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

类型

`object`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

类型

`string`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

类型

`object`

必填

`topologyKey`

属性	类型	描述
`labelSelector`	`object`	
`matchLabelKeys`	`array`	
`mismatchLabelKeys`	`array`	
`namespaceSelector`	`object`	
`namespaces`	`array`	
`topologyKey`	`string`	

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]

类型

`string`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels

类型

`object`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]

类型

`string`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]

类型

`string`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.requiredDuringSchedulingIgnoredDuring

Execution[].namespaceSelector.matchExpressions[].values

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]

类型

`string`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels

类型

`object`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]

类型

`string`

.spec.storage.storageClassDeviceSets[].preparePlacement.tolerations

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.tolerations[]

类型

`object`

属性	类型	描述
`effect`	`string`	
`key`	`string`	
`operator`	`string`	
`tolerationSeconds`	`integer`	
`value`	`string`	

.spec.storage.storageClassDeviceSets[].preparePlacement.topologySpreadConstraints

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.topologySpreadConstraints[]

类型

`object`

必填

`maxSkew` `topologyKey` `whenUnsatisfiable`

属性	类型	描述
`labelSelector`	`object`	
`matchLabelKeys`	`array`	
`maxSkew`	`integer`	
`minDomains`	`integer`	
`nodeAffinityPolicy`	`string`	
`nodeTaintsPolicy`	`string`	
`topologyKey`	`string`	
`whenUnsatisfiable`	`string`	

.spec.storage.storageClassDeviceSets[].preparePlacement.topologySpreadConstraints[].labelSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.storage.storageClassDeviceSets[].preparePlacement.topologySpreadConstraints[].labelSelector.matchExpressions

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.topologySpreadConstraints[].labelSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.storage.storageClassDeviceSets[].preparePlacement.topologySpreadConstraints[].labelSelector.matchExpres

sions[].values

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.topologySpreadConstraints[].labelSelector.matchExpressions[].values[]

类型

`string`

.spec.storage.storageClassDeviceSets[].preparePlacement.topologySpreadConstraints[].labelSelector.matchLabels

类型

`object`

.spec.storage.storageClassDeviceSets[].preparePlacement.topologySpreadConstraints[].matchLabelKeys

类型

`array`

.spec.storage.storageClassDeviceSets[].preparePlacement.topologySpreadConstraints[].matchLabelKeys[]

类型

`string`

.spec.storage.storageClassDeviceSets[].resources

描述

ResourceRequirements describes the compute resource requirements.

类型

`object`

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

.spec.storage.storageClassDeviceSets[].resources.claims

描述

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

类型

`array`

.spec.storage.storageClassDeviceSets[].resources.claims[]

描述

ResourceClaim references one entry in PodSpec.ResourceClaims.

类型

`object`

必填

`name`

属性	类型	描述
`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.storage.storageClassDeviceSets[].resources.limits

描述

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

类型

`object`

.spec.storage.storageClassDeviceSets[].resources.requests

描述

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/

类型

`object`

.spec.storage.storageClassDeviceSets[].volumeClaimTemplates

描述

VolumeClaimTemplates is a list of PVC templates for the underlying storage devices

类型

`array`

.spec.storage.storageClassDeviceSets[].volumeClaimTemplates[]

描述

VolumeClaimTemplate is a simplified version of K8s corev1's PVC. It has no type meta or status.

类型

`object`

属性	类型	描述
`metadata`	`ObjectMeta`	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata ↗
`spec`	`object`	spec defines the desired characteristics of a volume requested by a pod author. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims ↗

.spec.storage.storageClassDeviceSets[].volumeClaimTemplates[].spec

描述

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

类型

`object`

属性	类型	描述
`accessModes`	`array`	accessModes contains the desired access mod volume should have. More info: https://kubernetes.io/docs/concepts/storage/per volumes#access-modes-1 ↗
`dataSource`	`object`	dataSource field can be used to specify either:

属性	类型	描述
		• An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) • An existing PVC (PersistentVolumeClaim) If provisioner or an external controller can sup the specified data source, it will create a nev volume based on the contents of the specifie source. When the AnyVolumeDataSource fe gate is enabled, dataSource contents will be to dataSourceRef, and dataSourceRef conte be copied to dataSource when dataSourceRef.namespace is not specified. namespace is specified, then dataSourceRe not be copied to dataSource.
`dataSourceRef`	`object`	dataSourceRef specifies the object from which t populate the volume with data, if a non-empty v is desired. This may be any object from a non-e API group (non core object) or a PersistentVolumeClaim object. When this field i specified, volume binding will only succeed if th of the specified object matches some installed v populator or dynamic provisioner. This field will the functionality of the dataSource field and as s both fields are non-empty, they must have the s value. For backwards compatibility, when name isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to t same value automatically if one of them is empt the other is non-empty. When namespace is sp in dataSourceRef, dataSource isn't set to the sa value and must be empty. There are three impo differences between dataSource and dataSourc

属性	类型	描述
		• While dataSource only allows two specific ty objects, dataSourceRef allows any non-core as well as PersistentVolumeClaim objects. • While dataSource ignores disallowed values (dropping them), dataSourceRef preserves a values, and generates an error if a disallowe value is specified. • While dataSource only allows local objects, dataSourceRef allows objects in any names (Beta) Using this field requires the AnyVolumeDataSource feature gate to be en (Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource featur to be enabled.
`resources`	`object`	resources represents the minimum resources th volume should have. Users are allowed to spec resource requirements that are lower than previ value but must still be higher than capacity reco the status field of the claim. More info: https://kubernetes.io/docs/concepts/storage/per volumes#resources ↗
`selector`	`object`	selector is a label query over volumes to consid binding.

属性	类型	描述
`storageClassName`	`string`	storageClassName is the name of the StorageC required by the claim. More info: https://kubernetes.io/docs/concepts/storage/per volumes#class-1 ↗
`volumeAttributesClassName`	`string`	volumeAttributesClassName may be used to se VolumeAttributesClass used by this claim. If spe the CSI driver will create or update the volume v attributes defined in the corresponding VolumeAttributesClass. This has a different pur than storageClassName, it can be changed afte claim is created. An empty string or nil value inc that no VolumeAttributesClass will be applied to claim. If the claim enters an Infeasible error stat field can be reset to its previous value (including cancel the modification. If the resource referred volumeAttributesClass does not exist, this PersistentVolumeClaim will be set to a Pending as reflected by the modifyVolumeStatus field, u such as a resource exists. More info: https://kubernetes.io/docs/concepts/storage/vol attributes-classes/ ↗
`volumeMode`	`string`	volumeMode defines what type of volume is req by the claim. Value of Filesystem is implied whe included in claim spec.
`volumeName`	`string`	volumeName is the binding reference to the PersistentVolume backing this claim.

.spec.storage.storageClassDeviceSets[].volumeClaimTemplates[].spec.accessModes

描述

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

类型

`array`

.spec.storage.storageClassDeviceSets[].volumeClaimTemplates[].spec.accessModes[]

类型

`string`

.spec.storage.storageClassDeviceSets[].volumeClaimTemplates[].spec.dataSource

描述

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

类型

`object`

必填

`kind` `name`

属性	类型	描述
`apiGroup`	`string`	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
`kind`	`string`	Kind is the type of resource being referenced
`name`	`string`	Name is the name of resource being referenced

.spec.storage.storageClassDeviceSets[].volumeClaimTemplates[].spec.dataSourceRef

描述

dataSourceRef specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a PersistentVolumeClaim object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replace the functionality of the dataSource field and as such if both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in dataSourceRef, dataSource isn't set to the same value and must be empty. There are three important differences between dataSource and dataSourceRef: * While dataSource only allows two specific types of objects, dataSourceRef allows any non-core object, as well as PersistentVolumeClaim objects. * While dataSource ignores disallowed values (dropping them), dataSourceRef preserves all values, and generates an error if a disallowed value is specified. * While dataSource only allows local objects, dataSourceRef allows objects in any namespaces. (Beta) Using this field requires the AnyVolumeDataSource feature gate to be enabled.

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

类型

`object`

必填

`kind` `name`

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

.spec.storage.storageClassDeviceSets[].volumeClaimTemplates[].spec.resources

描述

resources represents the minimum resources the volume should have. Users are allowed to specify resource requirements that are lower than previous value but must still be higher than capacity recorded in the status field of the claim. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources

类型

`object`

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

.spec.storage.storageClassDeviceSets[].volumeClaimTemplates[].spec.resources.limits

描述

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

类型

`object`

.spec.storage.storageClassDeviceSets[].volumeClaimTemplates[].spec.resources.requests

描述

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/

类型

`object`

.spec.storage.storageClassDeviceSets[].volumeClaimTemplates[].spec.selector

描述

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

类型

`object`

属性	类型	描述
`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.storage.storageClassDeviceSets[].volumeClaimTemplates[].spec.selector.matchExpressions

描述

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

类型

`array`

.spec.storage.storageClassDeviceSets[].volumeClaimTemplates[].spec.selector.matchExpressions[]

描述

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

类型

`object`

必填

`key` `operator`

属性	类型	描述
`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.

属性	类型	描述
`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.storage.storageClassDeviceSets[].volumeClaimTemplates[].spec.selector.matchExpressions[].values

描述

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.

类型

`array`

.spec.storage.storageClassDeviceSets[].volumeClaimTemplates[].spec.selector.matchExpressions[].values[]

类型

`string`

.spec.storage.storageClassDeviceSets[].volumeClaimTemplates[].spec.selector.matchLabels

描述

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.

类型

`object`

.spec.storage.store

描述

OSDStore is the backend storage type used for creating the OSDs

类型

`object`

属性	类型	描述
`type`	`string`	Type of backend storage to be used while creating OSDs. If empty, then bluestore will be used
`updateStore`	`string`	UpdateStore updates the backend store for existing OSDs. It destroys each OSD one at a time, cleans up the backing disk and prepares same OSD on that disk

.spec.storage.volumeClaimTemplates

描述

PersistentVolumeClaims to use as storage

类型

`array`

.spec.storage.volumeClaimTemplates[]

描述

VolumeClaimTemplate is a simplified version of K8s corev1's PVC. It has no type meta or status.

类型

`object`

属性	类型	描述
`metadata`	`ObjectMeta`	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata ↗
`spec`	`object`	spec defines the desired characteristics of a volume requested by a pod author. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims ↗

.spec.storage.volumeClaimTemplates[].spec

描述

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

类型

`object`

属性	类型	描述
`accessModes`	`array`	accessModes contains the desired access mod volume should have. More info: https://kubernetes.io/docs/concepts/storage/per volumes#access-modes-1 ↗
`dataSource`	`object`	dataSource field can be used to specify either:

属性	类型	描述
		• An existing VolumeSnapshot object (snapshot.storage.k8s.io/VolumeSnapshot) • An existing PVC (PersistentVolumeClaim) If provisioner or an external controller can sup the specified data source, it will create a nev volume based on the contents of the specifie source. When the AnyVolumeDataSource fe gate is enabled, dataSource contents will be to dataSourceRef, and dataSourceRef conte be copied to dataSource when dataSourceRef.namespace is not specified. namespace is specified, then dataSourceRe not be copied to dataSource.
`dataSourceRef`	`object`	dataSourceRef specifies the object from which t populate the volume with data, if a non-empty v is desired. This may be any object from a non-e API group (non core object) or a PersistentVolumeClaim object. When this field i specified, volume binding will only succeed if th of the specified object matches some installed v populator or dynamic provisioner. This field will the functionality of the dataSource field and as s both fields are non-empty, they must have the s value. For backwards compatibility, when name isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to t same value automatically if one of them is empt the other is non-empty. When namespace is sp in dataSourceRef, dataSource isn't set to the sa value and must be empty. There are three impo differences between dataSource and dataSourc

属性	类型	描述
		• While dataSource only allows two specific ty objects, dataSourceRef allows any non-core as well as PersistentVolumeClaim objects. • While dataSource ignores disallowed values (dropping them), dataSourceRef preserves a values, and generates an error if a disallowe value is specified. • While dataSource only allows local objects, dataSourceRef allows objects in any names (Beta) Using this field requires the AnyVolumeDataSource feature gate to be en (Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource featur to be enabled.
`resources`	`object`	resources represents the minimum resources th volume should have. Users are allowed to spec resource requirements that are lower than previ value but must still be higher than capacity reco the status field of the claim. More info: https://kubernetes.io/docs/concepts/storage/per volumes#resources ↗
`selector`	`object`	selector is a label query over volumes to consid binding.

属性	类型	描述
`storageClassName`	`string`	storageClassName is the name of the StorageC required by the claim. More info: https://kubernetes.io/docs/concepts/storage/per volumes#class-1 ↗
`volumeAttributesClassName`	`string`	volumeAttributesClassName may be used to se VolumeAttributesClass used by this claim. If spe the CSI driver will create or update the volume v attributes defined in the corresponding VolumeAttributesClass. This has a different pur than storageClassName, it can be changed afte claim is created. An empty string or nil value inc that no VolumeAttributesClass will be applied to claim. If the claim enters an Infeasible error stat field can be reset to its previous value (including cancel the modification. If the resource referred volumeAttributesClass does not exist, this PersistentVolumeClaim will be set to a Pending as reflected by the modifyVolumeStatus field, u such as a resource exists. More info: https://kubernetes.io/docs/concepts/storage/vol attributes-classes/ ↗
`volumeMode`	`string`	volumeMode defines what type of volume is req by the claim. Value of Filesystem is implied whe included in claim spec.
`volumeName`	`string`	volumeName is the binding reference to the PersistentVolume backing this claim.

.spec.storage.volumeClaimTemplates[].spec.accessModes

描述

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

类型

`array`

.spec.storage.volumeClaimTemplates[].spec.accessModes[]

类型

`string`

.spec.storage.volumeClaimTemplates[].spec.dataSource

描述

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

类型

`object`

必填

`kind` `name`

属性	类型	描述
`apiGroup`	`string`	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
`kind`	`string`	Kind is the type of resource being referenced
`name`	`string`	Name is the name of resource being referenced

.spec.storage.volumeClaimTemplates[].spec.dataSourceRef

描述

dataSourceRef specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a PersistentVolumeClaim object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replace the functionality of the dataSource field and as such if both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in dataSourceRef, dataSource isn't set to the same value and must be empty. There are three important differences between dataSource and dataSourceRef: * While dataSource only allows two specific types of objects, dataSourceRef allows any non-core object, as well as PersistentVolumeClaim objects. * While dataSource ignores disallowed values (dropping them), dataSourceRef preserves all values, and generates an error if a disallowed value is specified. * While dataSource only allows local objects, dataSourceRef allows objects in any namespaces. (Beta) Using this field requires the AnyVolumeDataSource feature gate to be enabled.

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

类型

`object`

必填

`kind` `name`

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

.spec.storage.volumeClaimTemplates[].spec.resources

描述

resources represents the minimum resources the volume should have. Users are allowed to specify resource requirements that are lower than previous value but must still be higher than capacity recorded in the status field of the claim. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#resources

类型

`object`

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

.spec.storage.volumeClaimTemplates[].spec.resources.limits

描述

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

类型

`object`

.spec.storage.volumeClaimTemplates[].spec.resources.requests

描述

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/

类型

`object`

.spec.storage.volumeClaimTemplates[].spec.selector

描述

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

类型

`object`

属性	类型	描述
`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.storage.volumeClaimTemplates[].spec.selector.matchExpressions

描述

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

类型

`array`

.spec.storage.volumeClaimTemplates[].spec.selector.matchExpressions[]

描述

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

类型

`object`

必填

`key` `operator`

属性	类型	描述
`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.

属性	类型	描述
`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.storage.volumeClaimTemplates[].spec.selector.matchExpressions[].values

描述

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.

类型

`array`

.spec.storage.volumeClaimTemplates[].spec.selector.matchExpressions[].values[]

类型

`string`

.spec.storage.volumeClaimTemplates[].spec.selector.matchLabels

描述

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.

类型

`object`

.status

描述

ClusterStatus represents the status of a Ceph cluster

类型

`object`

属性	类型	描述
`ceph`	`object`	CephStatus is the details health of a Ceph Cluster
`cephx`	`object`	ClusterCephxStatus defines the cephx key rotation status of various daemons on the cephCluster resource
`conditions`	`array`	
`message`	`string`	
`observedGeneration`	`integer`	ObservedGeneration is the latest generation observed by the controller.
`phase`	`string`	ConditionType represent a resource's status
`state`	`string`	ClusterState represents the state of a Ceph Cluster

属性	类型	描述
`storage`	`object`	CephStorage represents flavors of Ceph Cluster Storage
`version`	`object`	ClusterVersion represents the version of a Ceph Cluster

.status.ceph

描述

CephStatus is the details health of a Ceph Cluster

类型

`object`

属性	类型	描述
`capacity`	`object`	Capacity is the capacity information of a Ceph Cluster
`details`	`object`	
`fsid`	`string`	
`health`	`string`	
`lastChanged`	`string`	
`lastChecked`	`string`	
`previousHealth`	`string`	

属性	类型	描述
`versions`	`object`	CephDaemonsVersions show the current ceph version for different ceph daemons

.status.ceph.capacity

描述

Capacity is the capacity information of a Ceph Cluster

类型

`object`

属性	类型	描述
`bytesAvailable`	`integer`	
`bytesTotal`	`integer`	
`bytesUsed`	`integer`	
`lastUpdated`	`string`	

.status.ceph.details

类型

`object`

.status.ceph.versions

描述

CephDaemonsVersions show the current ceph version for different ceph daemons

类型

`object`

属性	类型	描述
`cephfs-mirror`	`object`	CephFSMirror shows CephFSMirror Ceph version
`mds`	`object`	Mds shows Mds Ceph version
`mgr`	`object`	Mgr shows Mgr Ceph version
`mon`	`object`	Mon shows Mon Ceph version
`osd`	`object`	Osd shows Osd Ceph version
`overall`	`object`	Overall shows overall Ceph version
`rbd-mirror`	`object`	RbdMirror shows RbdMirror Ceph version
`rgw`	`object`	Rgw shows Rgw Ceph version

.status.ceph.versions.cephfs-mirror

描述

CephFSMirror shows CephFSMirror Ceph version

类型

`object`

.status.ceph.versions.mds

描述

Mds shows Mds Ceph version

类型

`object`

.status.ceph.versions.mgr

描述

Mgr shows Mgr Ceph version

类型

`object`

.status.ceph.versions.mon

描述

Mon shows Mon Ceph version

类型

`object`

.status.ceph.versions.osd

描述

Osd shows Osd Ceph version

类型

`object`

.status.ceph.versions.overall

描述

Overall shows overall Ceph version

类型

`object`

.status.ceph.versions.rbd-mirror

描述

RbdMirror shows RbdMirror Ceph version

类型

`object`

.status.ceph.versions.rgw

描述

Rgw shows Rgw Ceph version

类型

`object`

.status.cephx

描述

ClusterCephxStatus defines the cephx key rotation status of various daemons on the cephCluster resource

类型

`object`

属性	类型	描述
`admin`	`object`	Admin shows the CephX key status for the client.admin key

属性	类型	描述
`cephExporter`	`object`	Ceph Exporter represents the cephx key rotation status of the ceph exporter daemon
`crashCollector`	`object`	Crash Collector represents the cephx key rotation status of the crash collector daemon
`csi`	`object`	CSI shows the CephX key status for Ceph-CSI components.
`mgr`	`object`	Mgr represents the cephx key rotation status of the ceph manager daemon
`mon`	`object`	Mon represents the CephX key status of the Monitor daemons
`osd`	`object`	OSD shows the CephX key status of of OSDs
`rbdMirrorPeer`	`object`	RBDMirrorPeer represents the cephx key rotation status of the `rbd-mirror-peer` user

.status.cephx.admin

描述

Admin shows the CephX key status for the client.admin key

类型

`object`

属性	类型	描述
`keyCephVersion`	`string`	KeyCephVersion reports the Ceph version that created the current generation's keys. This is same string format as reported by `CephCluster.status.version.version` to allow them to be compared. E.g., `20.2.0-0`. For all newly-created resources, this field set to the version of Ceph that created the key. The special value "Uninitialized" indicates that keys are being created for the first time. An empty string indicates that the version is unknown, as expected in brownfield deployments.
`keyGeneration`	`integer`	KeyGeneration represents the CephX key generation for the last successful reconcile. For all newly-created resources, this field is set to `1`. When keys are rotated due to any rotation policy, the generation is incremented or updated to the configured policy generation. Generation `0` indicates that keys existed prior to the implementation of key tracking.

.status.cephx.cephExporter

描述

Ceph Exporter represents the cephx key rotation status of the ceph exporter daemon

类型

`object`

属性	类型	描述
`keyCephVersion`	`string`	KeyCephVersion reports the Ceph version that created the current generation's keys. This is same string format as reported by `CephCluster.status.version.version` to allow them to be compared. E.g., `20.2.0-0`. For all newly-created resources, this field set to the version of Ceph that created the key. The special value "Uninitialized" indicates that keys are being created for the first time. An empty string indicates that the version is unknown, as expected in brownfield deployments.
`keyGeneration`	`integer`	KeyGeneration represents the CephX key generation for the last successful reconcile. For all newly-created resources, this field is set to `1`. When keys are rotated due to any rotation policy, the generation is incremented or updated to the configured policy generation. Generation `0` indicates that keys existed prior to the implementation of key tracking.

.status.cephx.crashCollector

描述

Crash Collector represents the cephx key rotation status of the crash collector daemon

类型

`object`

属性	类型	描述
`keyCephVersion`	`string`	KeyCephVersion reports the Ceph version that created the current generation's keys. This is same string format as reported by `CephCluster.status.version.version` to allow them to be compared. E.g., `20.2.0-0`. For all newly-created resources, this field set to the version of Ceph that created the key. The special value "Uninitialized" indicates that keys are being created for the first time. An empty string indicates that the version is unknown, as expected in brownfield deployments.
`keyGeneration`	`integer`	KeyGeneration represents the CephX key generation for the last successful reconcile. For all newly-created resources, this field is set to `1`. When keys are rotated due to any rotation policy, the generation is incremented or updated to the configured policy generation. Generation `0` indicates that keys existed prior to the implementation of key tracking.

.status.cephx.csi

描述

CSI shows the CephX key status for Ceph-CSI components.

类型

`object`

属性	类型	描述
`keyCephVersion`	`string`	KeyCephVersion reports the Ceph version that created the current generation's keys. This is same string format as reported by `CephCluster.status.version.version` to allow them to be compared. E.g., `20.2.0-0`. For all newly-created resources, this field set to the version of Ceph that created the key. The special value "Uninitialized" indicates that keys are being created for the first time. An empty string indicates that the version is unknown, as expected in brownfield deployments.
`keyGeneration`	`integer`	KeyGeneration represents the CephX key generation for the last successful reconcile. For all newly-created resources, this field is set to `1`. When keys are rotated due to any rotation policy, the generation is incremented or updated to the configured policy generation. Generation `0` indicates that keys existed prior to the implementation of key tracking.
`priorKeyCount`	`integer`	PriorKeyCount reports the number of prior-generation CephX keys that remain active for the related component

.status.cephx.mgr

描述

Mgr represents the cephx key rotation status of the ceph manager daemon

类型

`object`

属性	类型	描述
`keyCephVersion`	`string`	KeyCephVersion reports the Ceph version that created the current generation's keys. This is same string format as reported by `CephCluster.status.version.version` to allow them to be compared. E.g., `20.2.0-0`. For all newly-created resources, this field set to the version of Ceph that created the key. The special value "Uninitialized" indicates that keys are being created for the first time. An empty string indicates that the version is unknown, as expected in brownfield deployments.
`keyGeneration`	`integer`	KeyGeneration represents the CephX key generation for the last successful reconcile. For all newly-created resources, this field is set to `1`. When keys are rotated due to any rotation policy, the generation is incremented or updated to the configured policy generation. Generation `0` indicates that keys existed prior to the implementation of key tracking.

.status.cephx.mon

描述

Mon represents the CephX key status of the Monitor daemons

类型

`object`

属性	类型	描述
`keyCephVersion`	`string`	KeyCephVersion reports the Ceph version that created the current generation's keys. This is same string format as reported by `CephCluster.status.version.version` to allow them to be compared. E.g., `20.2.0-0`. For all newly-created resources, this field set to the version of Ceph that created the key. The special value "Uninitialized" indicates that keys are being created for the first time. An empty string indicates that the version is unknown, as expected in brownfield deployments.
`keyGeneration`	`integer`	KeyGeneration represents the CephX key generation for the last successful reconcile. For all newly-created resources, this field is set to `1`. When keys are rotated due to any rotation policy, the generation is incremented or updated to the configured policy generation. Generation `0` indicates that keys existed prior to the implementation of key tracking.

.status.cephx.osd

描述

OSD shows the CephX key status of of OSDs

类型

`object`

属性	类型	描述
`keyCephVersion`	`string`	KeyCephVersion reports the Ceph version that created the current generation's keys. This is same string format as reported by `CephCluster.status.version.version` to allow them to be compared. E.g., `20.2.0-0`. For all newly-created resources, this field set to the version of Ceph that created the key. The special value "Uninitialized" indicates that keys are being created for the first time. An empty string indicates that the version is unknown, as expected in brownfield deployments.
`keyGeneration`	`integer`	KeyGeneration represents the CephX key generation for the last successful reconcile. For all newly-created resources, this field is set to `1`. When keys are rotated due to any rotation policy, the generation is incremented or updated to the configured policy generation. Generation `0` indicates that keys existed prior to the implementation of key tracking.

.status.cephx.rbdMirrorPeer

描述

RBDMirrorPeer represents the cephx key rotation status of the `rbd-mirror-peer` user

类型

`object`

属性	类型	描述
`keyCephVersion`	`string`	KeyCephVersion reports the Ceph version that created the current generation's keys. This is same string format as reported by `CephCluster.status.version.version` to allow them to be compared. E.g., `20.2.0-0`. For all newly-created resources, this field set to the version of Ceph that created the key. The special value "Uninitialized" indicates that keys are being created for the first time. An empty string indicates that the version is unknown, as expected in brownfield deployments.
`keyGeneration`	`integer`	KeyGeneration represents the CephX key generation for the last successful reconcile. For all newly-created resources, this field is set to `1`. When keys are rotated due to any rotation policy, the generation is incremented or updated to the configured policy generation. Generation `0` indicates that keys existed prior to the implementation of key tracking.

.status.conditions

类型

`array`

.status.conditions[]

描述

Condition represents a status condition on any Rook-Ceph Custom Resource.

类型

`object`

属性	类型	描述
`lastHeartbeatTime`	`string`	
`lastTransitionTime`	`string`	
`message`	`string`	
`reason`	`string`	ConditionReason is a reason for a condition
`status`	`string`	
`type`	`string`	ConditionType represent a resource's status

.status.storage

描述

CephStorage represents flavors of Ceph Cluster Storage

类型

`object`

属性	类型	描述
`deprecatedOSDs`	`object`	
`deviceClasses`	`array`	
`osd`	`object`	OSDStatus represents OSD status of the ceph Cluster

.status.storage.deprecatedOSDs

类型

`object`

.status.storage.deviceClasses

类型

`array`

.status.storage.deviceClasses[]

描述

DeviceClasses represents device classes of a Ceph Cluster

类型

`object`

属性	类型	描述
`name`	`string`	

.status.storage.osd

描述

OSDStatus represents OSD status of the ceph Cluster

类型

`object`

属性	类型	描述
`migrationStatus`	`object`	MigrationStatus status represents the current status of any OSD migration.

属性	类型	描述
`storeType`	`object`	StoreType is a mapping between the OSD backend stores and number of OSDs using these stores

.status.storage.osd.migrationStatus

描述

MigrationStatus status represents the current status of any OSD migration.

类型

`object`

属性	类型	描述
`pending`	`integer`	

.status.storage.osd.storeType

描述

StoreType is a mapping between the OSD backend stores and number of OSDs using these stores

类型

`object`

.status.version

描述

ClusterVersion represents the version of a Ceph Cluster

类型

`object`

属性	类型	描述
`image`	`string`	
`version`	`string`	

API 端点

可用的 API 端点如下：

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

/apis/ceph.rook.io/v1/namespaces/{namespace}/cephclusters

HTTP 方法

`DELETE`

描述

delete collection of CephCluster

HTTP 响应

HTTP 状态码	响应体
200 - OK	`Status` schema
401 - Unauthorized	Empty

HTTP 方法

`GET`

描述

list objects of kind CephCluster

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephClusterList` schema
401 - Unauthorized	Empty

HTTP 方法

`POST`

描述

create a new CephCluster

查询参数

参数	类型	描述
`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

参数	类型	描述
`fieldValidation`	`string`	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.

请求体参数

参数	类型	描述
`body`	`CephCluster` schema	`application/json` formatted

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephCluster` schema
201 - Created	`CephCluster` schema
202 - Accepted	`CephCluster` schema
401 - Unauthorized	Empty

/apis/ceph.rook.io/v1/namespaces/{namespace}/cephclusters/{name}

HTTP 方法

`DELETE`

描述

delete the specified CephCluster

查询参数

参数	类型	描述
`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 响应

HTTP 状态码	响应体
200 - OK	`Status` schema
202 - Accepted	`Status` schema
401 - Unauthorized	Empty

HTTP 方法

`GET`

描述

read the specified CephCluster

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephCluster` schema
401 - Unauthorized	Empty

HTTP 方法

`PATCH`

描述

partially update the specified CephCluster

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	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.

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephCluster` schema
401 - Unauthorized	Empty

HTTP 方法

`PUT`

描述

replace the specified CephCluster

查询参数

参数	类型	描述
`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

参数	类型	描述
`fieldValidation`	`string`	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.

请求体参数

参数	类型	描述
`body`	`CephCluster` schema	`application/json` formatted

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephCluster` schema
201 - Created	`CephCluster` schema
401 - Unauthorized	Empty

/apis/ceph.rook.io/v1/namespaces/{namespace}/cephclusters/{name}/status

HTTP 方法

`GET`

描述

read status of the specified CephCluster

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephCluster` schema
401 - Unauthorized	Empty

HTTP 方法

`PATCH`

描述

partially update status of the specified CephCluster

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	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.

参数	类型	描述
		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 响应

HTTP 状态码	响应体
200 - OK	`CephCluster` schema
401 - Unauthorized	Empty

HTTP 方法

`PUT`

描述

replace status of the specified CephCluster

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	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,

参数	类型	描述
		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.

请求体参数

参数	类型	描述
`body`	`CephCluster` schema	`application/json` formatted

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephCluster` schema
201 - Created	`CephCluster` schema
401 - Unauthorized	Empty

CephFilesystem [cephfilesystems.ceph.rook.io/v1]

描述

CephFilesystem represents a Ceph Filesystem

类型

`object`

必填

`metadata` `spec`

规格

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

属性	类型	描述
`kind`	`string`	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`metadata`	`ObjectMeta`	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
`spec`	`object`	FilesystemSpec represents the spec of a file system
`status`	`object`	CephFilesystemStatus represents the status of a Ceph Filesystem

.spec

描述

FilesystemSpec represents the spec of a file system

类型

`object`

必填

`dataPools` `metadataPool` `metadataServer`

属性	类型	描述
`dataPools`	`array`	The data pool settings, with optional predefined pool name.
`metadataPool`	`object`	The metadata pool settings
`metadataServer`	`object`	The mds pod info
`mirroring`	`object`	The mirroring settings
`preserveFilesystemOnDelete`	`boolean`	Preserve the fs in the cluster on CephFilesystem CR deletion. Setting this to true automatically implies PreservePoolsOnDelete is true.
`preservePoolNames`	`boolean`	Preserve pool names as specified
`preservePoolsOnDelete`	`boolean`	Preserve pools on filesystem deletion
`statusCheck`	`object`	The mirroring statusCheck

.spec.dataPools

描述

The data pool settings, with optional predefined pool name.

类型

`array`

.spec.dataPools[]

描述

NamedPoolSpec represents the named ceph pool spec

类型

`object`

属性	类型	描述
`application`	`string`	The application name to set on the pool. Only expected to be set for rgw pools.
`compressionMode`	`string`	DEPRECATED: use Parameters instead, e.g., Parameters["compression_mode"] = "force" The inline compression mode in Bluestore OSD to set to (options are: none, passive, aggressive, force) Do NOT set a default value for kubebuilder as this will override the Parameters
`crushRoot`	`string`	The root of the crush hierarchy utilized by the pool
`deviceClass`	`string`	The device class the OSD should set to for use in the pool

属性	类型	描述
`enableCrushUpdates`	`boolean`	Allow rook operator to change the pool CRUSH tunables once the pool is created
`enableRBDStats`	`boolean`	EnableRBDStats is used to enable gathering of statistics for all RBD images in the pool
`erasureCoded`	`object`	The erasure code settings
`failureDomain`	`string`	The failure domain: osd/host/(region or zone if available) - technically also any type in the crush map
`mirroring`	`object`	The mirroring settings
`name`	`string`	Name of the pool
`parameters`	`object`	Parameters is a list of properties to enable on a given pool
`quotas`	`object`	The quota settings
`replicated`	`object`	The replication settings

属性	类型	描述
`statusCheck`	`object`	The mirroring statusCheck

.spec.dataPools[].erasureCoded

描述

The erasure code settings

类型

`object`

必填

`codingChunks` `dataChunks`

属性	类型	描述
`algorithm`	`string`	The algorithm for erasure coding. If absent, defaults to the plugin specified in osd_pool_default_erasure_code_profile.
`codingChunks`	`integer`	Number of coding chunks per object in an erasure coded storage pool (required for erasure-coded pool type). This is the number of OSDs that can be lost simultaneously before data cannot be recovered.

属性	类型	描述
`dataChunks`	`integer`	Number of data chunks per object in an erasure coded storage pool (required for erasure-coded pool type). The number of chunks required to recover an object when any single OSD is lost is the same as dataChunks so be aware that the larger the number of data chunks, the higher the cost of recovery.

.spec.dataPools[].mirroring

描述

The mirroring settings

类型

`object`

属性	类型	描述
`enabled`	`boolean`	Enabled whether this pool is mirrored or not
`mode`	`string`	Mode is the mirroring mode: pool, image or init-only.
`peers`	`object`	Peers represents the peers spec
`snapshotSchedules`	`array`	SnapshotSchedules is the scheduling of snapshot for mirrored images/pools

.spec.dataPools[].mirroring.peers

描述

Peers represents the peers spec

类型

`object`

属性	类型	描述
`secretNames`	`array`	SecretNames represents the Kubernetes Secret names to add rbd-mirror or cephfs-mirror peers

.spec.dataPools[].mirroring.peers.secretNames

描述

SecretNames represents the Kubernetes Secret names to add rbd-mirror or cephfs-mirror peers

类型

`array`

.spec.dataPools[].mirroring.peers.secretNames[]

类型

`string`

.spec.dataPools[].mirroring.snapshotSchedules

描述

SnapshotSchedules is the scheduling of snapshot for mirrored images/pools

类型

`array`

.spec.dataPools[].mirroring.snapshotSchedules[]

描述

SnapshotScheduleSpec represents the snapshot scheduling settings of a mirrored pool

类型

`object`

属性	类型	描述
`interval`	`string`	Interval represent the periodicity of the snapshot.
`path`	`string`	Path is the path to snapshot, only valid for CephFS
`startTime`	`string`	StartTime indicates when to start the snapshot

.spec.dataPools[].parameters

描述

Parameters is a list of properties to enable on a given pool

类型

`object`

.spec.dataPools[].quotas

描述

The quota settings

类型

`object`

属性	类型	描述
`maxBytes`	`integer`	MaxBytes represents the quota in bytes Deprecated in favor of MaxSize
`maxObjects`	`integer`	MaxObjects represents the quota in objects
`maxSize`	`string`	MaxSize represents the quota in bytes as a string

.spec.dataPools[].replicated

描述

The replication settings

类型

`object`

必填

`size`

属性	类型	描述
`hybridStorage`	`object`	HybridStorage represents hybrid storage tier settings
`replicasPerFailureDomain`	`integer`	ReplicasPerFailureDomain the number of replica in the specified failure domain

属性	类型	描述
`requireSafeReplicaSize`	`boolean`	RequireSafeReplicaSize if false allows you to set replica 1
`size`	`integer`	Size - Number of copies per object in a replicated storage pool, including the object itself (required for replicated pool type)
`subFailureDomain`	`string`	SubFailureDomain the name of the sub-failure domain
`targetSizeRatio`	`number`	TargetSizeRatio gives a hint (%) to Ceph in terms of expected consumption of the total cluster capacity

.spec.dataPools[].replicated.hybridStorage

描述

HybridStorage represents hybrid storage tier settings

类型

`object`

必填

`primaryDeviceClass` `secondaryDeviceClass`

属性	类型	描述
`primaryDeviceClass`	`string`	PrimaryDeviceClass represents high performance tier (for example SSD or NVME) for Primary OSD
`secondaryDeviceClass`	`string`	SecondaryDeviceClass represents low performance tier (for example HDDs) for remaining OSDs

.spec.dataPools[].statusCheck

描述

The mirroring statusCheck

类型

`object`

属性	类型	描述
`mirror`	`object`	HealthCheckSpec represents the health check of an object store bucket

.spec.dataPools[].statusCheck.mirror

描述

HealthCheckSpec represents the health check of an object store bucket

类型

`object`

属性	类型	描述
`disabled`	`boolean`	
`interval`	`string`	Interval is the internal in second or minute for the health check to run like 60s for 60 seconds
`timeout`	`string`	

.spec.metadataPool

描述

The metadata pool settings

类型

`object`

属性	类型	描述
`application`	`string`	The application name to set on the pool. Only expected to be set for rgw pools.
`compressionMode`	`string`	DEPRECATED: use Parameters instead, e.g., Parameters["compression_mode"] = "force" The inline compression mode in Bluestore OSD to set to (options are: none, passive, aggressive, force) Do NOT set a default value for kubebuilder as this will override the Parameters
`crushRoot`	`string`	The root of the crush hierarchy utilized by the pool

属性	类型	描述
`deviceClass`	`string`	The device class the OSD should set to for use in the pool
`enableCrushUpdates`	`boolean`	Allow rook operator to change the pool CRUSH tunables once the pool is created
`enableRBDStats`	`boolean`	EnableRBDStats is used to enable gathering of statistics for all RBD images in the pool
`erasureCoded`	`object`	The erasure code settings
`failureDomain`	`string`	The failure domain: osd/host/(region or zone if available) - technically also any type in the crush map
`mirroring`	`object`	The mirroring settings
`name`	`string`	Name of the pool
`parameters`	`object`	Parameters is a list of properties to enable on a given pool

属性	类型	描述
`quotas`	`object`	The quota settings
`replicated`	`object`	The replication settings
`statusCheck`	`object`	The mirroring statusCheck

.spec.metadataPool.erasureCoded

描述

The erasure code settings

类型

`object`

必填

`codingChunks` `dataChunks`

属性	类型	描述
`algorithm`	`string`	The algorithm for erasure coding. If absent, defaults to the plugin specified in osd_pool_default_erasure_code_profile.
`codingChunks`	`integer`	Number of coding chunks per object in an erasure coded storage pool (required for erasure-coded pool type). This is the number of OSDs that can be lost simultaneously before data cannot be recovered.

属性	类型	描述
`dataChunks`	`integer`	Number of data chunks per object in an erasure coded storage pool (required for erasure-coded pool type). The number of chunks required to recover an object when any single OSD is lost is the same as dataChunks so be aware that the larger the number of data chunks, the higher the cost of recovery.

.spec.metadataPool.mirroring

描述

The mirroring settings

类型

`object`

属性	类型	描述
`enabled`	`boolean`	Enabled whether this pool is mirrored or not
`mode`	`string`	Mode is the mirroring mode: pool, image or init-only.
`peers`	`object`	Peers represents the peers spec
`snapshotSchedules`	`array`	SnapshotSchedules is the scheduling of snapshot for mirrored images/pools

.spec.metadataPool.mirroring.peers

描述

Peers represents the peers spec

类型

`object`

属性	类型	描述
`secretNames`	`array`	SecretNames represents the Kubernetes Secret names to add rbd-mirror or cephfs-mirror peers

.spec.metadataPool.mirroring.peers.secretNames

描述

SecretNames represents the Kubernetes Secret names to add rbd-mirror or cephfs-mirror peers

类型

`array`

.spec.metadataPool.mirroring.peers.secretNames[]

类型

`string`

.spec.metadataPool.mirroring.snapshotSchedules

描述

SnapshotSchedules is the scheduling of snapshot for mirrored images/pools

类型

`array`

.spec.metadataPool.mirroring.snapshotSchedules[]

描述

SnapshotScheduleSpec represents the snapshot scheduling settings of a mirrored pool

类型

`object`

属性	类型	描述
`interval`	`string`	Interval represent the periodicity of the snapshot.
`path`	`string`	Path is the path to snapshot, only valid for CephFS
`startTime`	`string`	StartTime indicates when to start the snapshot

.spec.metadataPool.parameters

描述

Parameters is a list of properties to enable on a given pool

类型

`object`

.spec.metadataPool.quotas

描述

The quota settings

类型

`object`

属性	类型	描述
`maxBytes`	`integer`	MaxBytes represents the quota in bytes Deprecated in favor of MaxSize
`maxObjects`	`integer`	MaxObjects represents the quota in objects
`maxSize`	`string`	MaxSize represents the quota in bytes as a string

.spec.metadataPool.replicated

描述

The replication settings

类型

`object`

必填

`size`

属性	类型	描述
`hybridStorage`	`object`	HybridStorage represents hybrid storage tier settings
`replicasPerFailureDomain`	`integer`	ReplicasPerFailureDomain the number of replica in the specified failure domain

属性	类型	描述
`requireSafeReplicaSize`	`boolean`	RequireSafeReplicaSize if false allows you to set replica 1
`size`	`integer`	Size - Number of copies per object in a replicated storage pool, including the object itself (required for replicated pool type)
`subFailureDomain`	`string`	SubFailureDomain the name of the sub-failure domain
`targetSizeRatio`	`number`	TargetSizeRatio gives a hint (%) to Ceph in terms of expected consumption of the total cluster capacity

.spec.metadataPool.replicated.hybridStorage

描述

HybridStorage represents hybrid storage tier settings

类型

`object`

必填

`primaryDeviceClass` `secondaryDeviceClass`

属性	类型	描述
`primaryDeviceClass`	`string`	PrimaryDeviceClass represents high performance tier (for example SSD or NVME) for Primary OSD
`secondaryDeviceClass`	`string`	SecondaryDeviceClass represents low performance tier (for example HDDs) for remaining OSDs

.spec.metadataPool.statusCheck

描述

The mirroring statusCheck

类型

`object`

属性	类型	描述
`mirror`	`object`	HealthCheckSpec represents the health check of an object store bucket

.spec.metadataPool.statusCheck.mirror

描述

HealthCheckSpec represents the health check of an object store bucket

类型

`object`

属性	类型	描述
`disabled`	`boolean`	
`interval`	`string`	Interval is the internal in second or minute for the health check to run like 60s for 60 seconds
`timeout`	`string`	

.spec.metadataServer

描述

The mds pod info

类型

`object`

必填

`activeCount`

属性	类型	描述
`activeCount`	`integer`	The number of metadata servers that are active. The remaining servers in the cluster will be in standby mode.
`activeStandby`	`boolean`	Whether each active MDS instance will have an active standby with a warm metadata cache for faster failover. If false, standbys will still be available, but will not have a warm metadata cache.

属性	类型	描述
`annotations`	`object`	The annotations-related configuration to add/set on each Pod related object.
`cacheMemoryLimitFactor`	`number`	CacheMemoryLimitFactor is the factor applied to the memory limit to determine the MDS cache memory limit. MDS cache memory limit should be set to 50-60% of RAM reserved for the MDS container. MDS uses approximately 125% of the value of mds_cache_memory_limit in RAM. This factor is applied when resources.limits.memory is set.
`cacheMemoryRequestFactor`	`number`	CacheMemoryRequestFactor is the factor applied to the memory request to determine the MDS cache memory limit. This factor is applied when resources.requests.memory is set and resources.limits.memory is not set.
`labels`	`object`	The labels-related configuration to add/set on each Pod related object.
`livenessProbe`	`object`	ProbeSpec is a wrapper around Probe so it can be enabled or disabled for a Ceph daemon
`placement`	`object`	

属性	类型	描述
`priorityClassName`	`string`	PriorityClassName sets priority classes on components
`resources`	`object`	The resource requirements for the mds pods
`startupProbe`	`object`	ProbeSpec is a wrapper around Probe so it can be enabled or disabled for a Ceph daemon

.spec.metadataServer.annotations

描述

The annotations-related configuration to add/set on each Pod related object.

类型

`object`

.spec.metadataServer.labels

描述

The labels-related configuration to add/set on each Pod related object.

类型

`object`

.spec.metadataServer.livenessProbe

描述

ProbeSpec is a wrapper around Probe so it can be enabled or disabled for a Ceph daemon

类型

`object`

属性	类型	描述
`disabled`	`boolean`	Disabled determines whether probe is disable or not
`probe`	`object`	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

.spec.metadataServer.livenessProbe.probe

描述

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

类型

`object`

属性	类型	描述
`exec`	`object`	Exec specifies a command to execute in th
`failureThreshold`	`integer`	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.

属性	类型	描述
`grpc`	`object`	GRPC specifies a GRPC HealthCheckReq
`httpGet`	`object`	HTTPGet specifies an HTTP GET request
`initialDelaySeconds`	`integer`	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗
`periodSeconds`	`integer`	How often (in seconds) to perform the prob 10 seconds. Minimum value is 1.
`successThreshold`	`integer`	Minimum consecutive successes for the pr considered successful after having failed. I Must be 1 for liveness and startup. Minimu
`tcpSocket`	`object`	TCPSocket specifies a connection to a TC
`terminationGracePeriodSeconds`	`integer`	
`timeoutSeconds`	`integer`	Number of seconds after which the probe t Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗

.spec.metadataServer.livenessProbe.probe.exec

描述

Exec specifies a command to execute in the container.

类型

`object`

属性	类型	描述
`command`	`array`	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.metadataServer.livenessProbe.probe.exec.command

描述

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

类型

`array`

.spec.metadataServer.livenessProbe.probe.exec.command[]

类型

`string`

.spec.metadataServer.livenessProbe.probe.grpc

描述

GRPC specifies a GRPC HealthCheckRequest.

类型

`object`

必填

`port`

属性	类型	描述
`port`	`integer`	Port number of the gRPC service. Number must be in the range 1 to 65535.
`service`	`string`	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md↗). If this is not specified, the default behavior is defined by gRPC.

.spec.metadataServer.livenessProbe.probe.httpGet

描述

HTTPGet specifies an HTTP GET request to perform.

类型

`object`

必填

`port`

属性	类型	描述
`host`	`string`	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
`httpHeaders`	`array`	Custom headers to set in the request. HTTP allows repeated headers.
`path`	`string`	Path to access on the HTTP server.
`port`	`integer|string`	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
`scheme`	`string`	Scheme to use for connecting to the host. Defaults to HTTP.

.spec.metadataServer.livenessProbe.probe.httpGet.httpHeaders

描述

Custom headers to set in the request. HTTP allows repeated headers.

类型

`array`

.spec.metadataServer.livenessProbe.probe.httpGet.httpHeaders[]

描述

HTTPHeader describes a custom header to be used in HTTP probes

类型

`object`

必填

`name` `value`

属性	类型	描述
`name`	`string`	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
`value`	`string`	The header field value

.spec.metadataServer.livenessProbe.probe.tcpSocket

描述

TCPSocket specifies a connection to a TCP port.

类型

`object`

必填

`port`

属性	类型	描述
`host`	`string`	Optional: Host name to connect to, defaults to the pod IP.
`port`	`integer|string`	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

.spec.metadataServer.placement

类型

`object`

属性	类型	描述
`nodeAffinity`	`object`	
`podAffinity`	`object`	
`podAntiAffinity`	`object`	
`tolerations`	`array`	
`topologySpreadConstraints`	`array`	

.spec.metadataServer.placement.nodeAffinity

类型

`object`

属性	类型	描述
`preferredDuringSchedulingIgnoredDuringExecution`	`array`	
`requiredDuringSchedulingIgnoredDuringExecution`	`object`	

.spec.metadataServer.placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution

类型

`array`

.spec.metadataServer.placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

类型

`object`

必填

`preference` `weight`

属性	类型	描述
`preference`	`object`	
`weight`	`integer`	

.spec.metadataServer.placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchFields`	`array`	

.spec.metadataServer.placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions

类型

`array`

.spec.metadataServer.placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.metadataServer.placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matc

hExpressions[].values

类型

`array`

.spec.metadataServer.placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values[]

类型

`string`

.spec.metadataServer.placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields

类型

`array`

.spec.metadataServer.placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	

属性	类型	描述
`operator`	`string`	
`values`	`array`	

.spec.metadataServer.placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values

类型

`array`

.spec.metadataServer.placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values[]

类型

`string`

.spec.metadataServer.placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution

类型

`object`

必填

`nodeSelectorTerms`

属性	类型	描述
`nodeSelectorTerms`	`array`	

.spec.metadataServer.placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms

类型

`array`

.spec.metadataServer.placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[]

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchFields`	`array`	

.spec.metadataServer.placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions

类型

`array`

.spec.metadataServer.placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.metadataServer.placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values

类型

`array`

.spec.metadataServer.placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values[]

类型

`string`

.spec.metadataServer.placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields

类型

`array`

.spec.metadataServer.placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.metadataServer.placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values

类型

`array`

.spec.metadataServer.placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values[]

类型

`string`

.spec.metadataServer.placement.podAffinity

类型

`object`

属性	类型	描述
`preferredDuringSchedulingIgnoredDuringExecution`	`array`	
`requiredDuringSchedulingIgnoredDuringExecution`	`array`	

.spec.metadataServer.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution

类型

`array`

.spec.metadataServer.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

类型

`object`

必填

`podAffinityTerm` `weight`

属性	类型	描述
`podAffinityTerm`	`object`	
`weight`	`integer`	

.spec.metadataServer.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm

类型

`object`

必填

`topologyKey`

属性	类型	描述
`labelSelector`	`object`	
`matchLabelKeys`	`array`	
`mismatchLabelKeys`	`array`	
`namespaceSelector`	`object`	
`namespaces`	`array`	
`topologyKey`	`string`	

.spec.metadataServer.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.metadataServer.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions

类型

`array`

.spec.metadataServer.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.metadataServer.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

类型

`array`

.spec.metadataServer.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]

类型

`string`

.spec.metadataServer.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels

类型

`object`

.spec.metadataServer.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys

类型

`array`

.spec.metadataServer.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]

类型

`string`

.spec.metadataServer.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys

类型

`array`

.spec.metadataServer.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]

类型

`string`

.spec.metadataServer.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.metadataServer.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions

类型

`array`

.spec.metadataServer.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.metadataServer.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values

类型

`array`

.spec.metadataServer.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]

类型

`string`

.spec.metadataServer.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

类型

`object`

.spec.metadataServer.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

类型

`array`

.spec.metadataServer.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

类型

`string`

.spec.metadataServer.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution

类型

`array`

.spec.metadataServer.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

类型

`object`

必填

`topologyKey`

属性	类型	描述
`labelSelector`	`object`	
`matchLabelKeys`	`array`	
`mismatchLabelKeys`	`array`	
`namespaceSelector`	`object`	
`namespaces`	`array`	
`topologyKey`	`string`	

.spec.metadataServer.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.metadataServer.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions

类型

`array`

.spec.metadataServer.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.metadataServer.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values

类型

`array`

.spec.metadataServer.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]

类型

`string`

.spec.metadataServer.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels

类型

`object`

.spec.metadataServer.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys

类型

`array`

.spec.metadataServer.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]

类型

`string`

.spec.metadataServer.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKe

ys

类型

`array`

.spec.metadataServer.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]

类型

`string`

.spec.metadataServer.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.metadataServer.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions

类型

`array`

.spec.metadataServer.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.metadataServer.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

类型

`array`

.spec.metadataServer.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]

类型

`string`

.spec.metadataServer.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels

类型

`object`

.spec.metadataServer.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces

类型

`array`

.spec.metadataServer.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]

类型

`string`

.spec.metadataServer.placement.podAntiAffinity

类型

`object`

属性	类型	描述
`preferredDuringSchedulingIgnoredDuringExecution`	`array`	
`requiredDuringSchedulingIgnoredDuringExecution`	`array`	

.spec.metadataServer.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution

类型

`array`

.spec.metadataServer.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

类型

`object`

必填

`podAffinityTerm` `weight`

属性	类型	描述
`podAffinityTerm`	`object`	
`weight`	`integer`	

.spec.metadataServer.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm

类型

`object`

必填

`topologyKey`

属性	类型	描述
`labelSelector`	`object`	

属性	类型	描述
`matchLabelKeys`	`array`	
`mismatchLabelKeys`	`array`	
`namespaceSelector`	`object`	
`namespaces`	`array`	
`topologyKey`	`string`	

.spec.metadataServer.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.metadataServer.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions

类型

`array`

.spec.metadataServer.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.metadataServer.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

类型

`array`

.spec.metadataServer.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]

类型

`string`

.spec.metadataServer.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels

类型

`object`

.spec.metadataServer.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys

类型

`array`

.spec.metadataServer.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]

类型

`string`

.spec.metadataServer.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys

类型

`array`

.spec.metadataServer.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]

类型

`string`

.spec.metadataServer.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.metadataServer.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions

类型

`array`

.spec.metadataServer.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.metadataServer.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values

类型

`array`

.spec.metadataServer.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]

类型

`string`

.spec.metadataServer.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

类型

`object`

.spec.metadataServer.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

类型

`array`

.spec.metadataServer.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

类型

`string`

.spec.metadataServer.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution

类型

`array`

.spec.metadataServer.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

类型

`object`

必填

`topologyKey`

属性	类型	描述
`labelSelector`	`object`	
`matchLabelKeys`	`array`	
`mismatchLabelKeys`	`array`	
`namespaceSelector`	`object`	
`namespaces`	`array`	
`topologyKey`	`string`	

.spec.metadataServer.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.metadataServer.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions

类型

`array`

.spec.metadataServer.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.metadataServer.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values

类型

`array`

.spec.metadataServer.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]

类型

`string`

.spec.metadataServer.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels

类型

`object`

.spec.metadataServer.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys

类型

`array`

.spec.metadataServer.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]

类型

`string`

.spec.metadataServer.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys

类型

`array`

.spec.metadataServer.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]

类型

`string`

.spec.metadataServer.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.metadataServer.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions

类型

`array`

.spec.metadataServer.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.metadataServer.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

类型

`array`

.spec.metadataServer.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]

类型

`string`

.spec.metadataServer.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels

类型

`object`

.spec.metadataServer.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces

类型

`array`

.spec.metadataServer.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]

类型

`string`

.spec.metadataServer.placement.tolerations

类型

`array`

.spec.metadataServer.placement.tolerations[]

类型

`object`

属性	类型	描述
`effect`	`string`	
`key`	`string`	
`operator`	`string`	

属性	类型	描述
`tolerationSeconds`	`integer`	
`value`	`string`	

.spec.metadataServer.placement.topologySpreadConstraints

类型

`array`

.spec.metadataServer.placement.topologySpreadConstraints[]

类型

`object`

必填

`maxSkew` `topologyKey` `whenUnsatisfiable`

属性	类型	描述
`labelSelector`	`object`	
`matchLabelKeys`	`array`	
`maxSkew`	`integer`	
`minDomains`	`integer`	
`nodeAffinityPolicy`	`string`	
`nodeTaintsPolicy`	`string`	
`topologyKey`	`string`	

属性	类型	描述
`whenUnsatisfiable`	`string`	

.spec.metadataServer.placement.topologySpreadConstraints[].labelSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.metadataServer.placement.topologySpreadConstraints[].labelSelector.matchExpressions

类型

`array`

.spec.metadataServer.placement.topologySpreadConstraints[].labelSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.metadataServer.placement.topologySpreadConstraints[].labelSelector.matchExpressions[].values

类型

`array`

.spec.metadataServer.placement.topologySpreadConstraints[].labelSelector.matchExpressions[].values[]

类型

`string`

.spec.metadataServer.placement.topologySpreadConstraints[].labelSelector.matchLabels

类型

`object`

.spec.metadataServer.placement.topologySpreadConstraints[].matchLabelKeys

类型

`array`

.spec.metadataServer.placement.topologySpreadConstraints[].matchLabelKeys[]

类型

`string`

.spec.metadataServer.resources

描述

The resource requirements for the mds pods

类型

`object`

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

属性	类型	描述
		Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

.spec.metadataServer.resources.claims

描述

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

类型

`array`

.spec.metadataServer.resources.claims[]

描述

ResourceClaim references one entry in PodSpec.ResourceClaims.

类型

`object`

必填

`name`

属性	类型	描述
`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.

属性	类型	描述
`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.metadataServer.resources.limits

描述

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

类型

`object`

.spec.metadataServer.resources.requests

描述

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/

类型

`object`

.spec.metadataServer.startupProbe

描述

ProbeSpec is a wrapper around Probe so it can be enabled or disabled for a Ceph daemon

类型

`object`

属性	类型	描述
`disabled`	`boolean`	Disabled determines whether probe is disable or not
`probe`	`object`	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

.spec.metadataServer.startupProbe.probe

描述

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

类型

`object`

属性	类型	描述
`exec`	`object`	Exec specifies a command to execute in th
`failureThreshold`	`integer`	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
`grpc`	`object`	GRPC specifies a GRPC HealthCheckReq
`httpGet`	`object`	HTTPGet specifies an HTTP GET request

属性	类型	描述
`initialDelaySeconds`	`integer`	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗
`periodSeconds`	`integer`	How often (in seconds) to perform the prob 10 seconds. Minimum value is 1.
`successThreshold`	`integer`	Minimum consecutive successes for the pr considered successful after having failed. I Must be 1 for liveness and startup. Minimu
`tcpSocket`	`object`	TCPSocket specifies a connection to a TC
`terminationGracePeriodSeconds`	`integer`	
`timeoutSeconds`	`integer`	Number of seconds after which the probe t Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗

.spec.metadataServer.startupProbe.probe.exec

描述

Exec specifies a command to execute in the container.

类型

`object`

属性	类型	描述
`command`	`array`	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.metadataServer.startupProbe.probe.exec.command

描述

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

类型

`array`

.spec.metadataServer.startupProbe.probe.exec.command[ ]

类型

`string`

.spec.metadataServer.startupProbe.probe.grpc

描述

GRPC specifies a GRPC HealthCheckRequest.

类型

`object`

必填

`port`

属性	类型	描述
`port`	`integer`	Port number of the gRPC service. Number must be in the range 1 to 65535.
`service`	`string`	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md↗). If this is not specified, the default behavior is defined by gRPC.

.spec.metadataServer.startupProbe.probe.httpGet

描述

HTTPGet specifies an HTTP GET request to perform.

类型

`object`

必填

`port`

属性	类型	描述
`host`	`string`	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
`httpHeaders`	`array`	Custom headers to set in the request. HTTP allows repeated headers.
`path`	`string`	Path to access on the HTTP server.
`port`	`integer|string`	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
`scheme`	`string`	Scheme to use for connecting to the host. Defaults to HTTP.

.spec.metadataServer.startupProbe.probe.httpGet.httpHeaders

描述

Custom headers to set in the request. HTTP allows repeated headers.

类型

`array`

.spec.metadataServer.startupProbe.probe.httpGet.httpHeaders[]

描述

HTTPHeader describes a custom header to be used in HTTP probes

类型

`object`

必填

`name` `value`

属性	类型	描述
`name`	`string`	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
`value`	`string`	The header field value

.spec.metadataServer.startupProbe.probe.tcpSocket

描述

TCPSocket specifies a connection to a TCP port.

类型

`object`

必填

`port`

属性	类型	描述
`host`	`string`	Optional: Host name to connect to, defaults to the pod IP.

属性	类型	描述
`port`	`integer|string`	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

.spec.mirroring

描述

The mirroring settings

类型

`object`

属性	类型	描述
`enabled`	`boolean`	Enabled whether this filesystem is mirrored or not
`peers`	`object`	Peers represents the peers spec
`snapshotRetention`	`array`	Retention is the retention policy for a snapshot schedule One path has exactly one retention policy. A policy can however contain multiple count-time period pairs in order to specify complex retention policies
`snapshotSchedules`	`array`	SnapshotSchedules is the scheduling of snapshot for mirrored filesystems

.spec.mirroring.peers

描述

Peers represents the peers spec

类型

`object`

属性	类型	描述
`secretNames`	`array`	SecretNames represents the Kubernetes Secret names to add rbd-mirror or cephfs-mirror peers

.spec.mirroring.peers.secretNames

描述

SecretNames represents the Kubernetes Secret names to add rbd-mirror or cephfs-mirror peers

类型

`array`

.spec.mirroring.peers.secretNames[]

类型

`string`

.spec.mirroring.snapshotRetention

描述

Retention is the retention policy for a snapshot schedule One path has exactly one retention policy. A policy can however contain multiple count-time period pairs in order to specify complex retention policies

类型

`array`

.spec.mirroring.snapshotRetention[]

描述

SnapshotScheduleRetentionSpec is a retention policy

类型

`object`

属性	类型	描述
`duration`	`string`	Duration represents the retention duration for a snapshot
`path`	`string`	Path is the path to snapshot

.spec.mirroring.snapshotSchedules

描述

SnapshotSchedules is the scheduling of snapshot for mirrored filesystems

类型

`array`

.spec.mirroring.snapshotSchedules[]

描述

SnapshotScheduleSpec represents the snapshot scheduling settings of a mirrored pool

类型

`object`

属性	类型	描述
`interval`	`string`	Interval represent the periodicity of the snapshot.
`path`	`string`	Path is the path to snapshot, only valid for CephFS
`startTime`	`string`	StartTime indicates when to start the snapshot

.spec.statusCheck

描述

The mirroring statusCheck

类型

`object`

属性	类型	描述
`mirror`	`object`	HealthCheckSpec represents the health check of an object store bucket

.spec.statusCheck.mirror

描述

HealthCheckSpec represents the health check of an object store bucket

类型

`object`

属性	类型	描述
`disabled`	`boolean`	
`interval`	`string`	Interval is the internal in second or minute for the health check to run like 60s for 60 seconds
`timeout`	`string`	

.status

描述

CephFilesystemStatus represents the status of a Ceph Filesystem

类型

`object`

属性	类型	描述
`cephx`	`object`	
`conditions`	`array`	
`info`	`object`	Use only info and put mirroringStatus in it?
`mirroringStatus`	`object`	MirroringStatus is the filesystem mirroring status
`observedGeneration`	`integer`	ObservedGeneration is the latest generation observed by the controller.

属性	类型	描述
`phase`	`string`	ConditionType represent a resource's status
`snapshotScheduleStatus`	`object`	FilesystemSnapshotScheduleStatusSpec is the status of the snapshot schedule

.status.cephx

类型

`object`

属性	类型	描述
`daemon`	`object`	Daemon shows the CephX key status for local Ceph daemons associated with this resources.

.status.cephx.daemon

描述

Daemon shows the CephX key status for local Ceph daemons associated with this resources.

类型

`object`

属性	类型	描述
`keyCephVersion`	`string`	KeyCephVersion reports the Ceph version that created the current generation's keys. This is same string

属性	类型	描述
		format as reported by `CephCluster.status.version.version` to allow them to be compared. E.g., `20.2.0-0`. For all newly-created resources, this field set to the version of Ceph that created the key. The special value "Uninitialized" indicates that keys are being created for the first time. An empty string indicates that the version is unknown, as expected in brownfield deployments.
`keyGeneration`	`integer`	KeyGeneration represents the CephX key generation for the last successful reconcile. For all newly-created resources, this field is set to `1`. When keys are rotated due to any rotation policy, the generation is incremented or updated to the configured policy generation. Generation `0` indicates that keys existed prior to the implementation of key tracking.

.status.conditions

类型

`array`

.status.conditions[]

描述

Condition represents a status condition on any Rook-Ceph Custom Resource.

类型

`object`

属性	类型	描述
`lastHeartbeatTime`	`string`	
`lastTransitionTime`	`string`	
`message`	`string`	
`reason`	`string`	ConditionReason is a reason for a condition
`status`	`string`	
`type`	`string`	ConditionType represent a resource's status

.status.info

描述

Use only info and put mirroringStatus in it?

类型

`object`

.status.mirroringStatus

描述

MirroringStatus is the filesystem mirroring status

类型

`object`

属性	类型	描述
`daemonsStatus`	`array`	PoolMirroringStatus is the mirroring status of a filesystem
`details`	`string`	Details contains potential status errors
`lastChanged`	`string`	LastChanged is the last time time the status last changed
`lastChecked`	`string`	LastChecked is the last time time the status was checked

.status.mirroringStatus.daemonsStatus

描述

PoolMirroringStatus is the mirroring status of a filesystem

类型

`array`

.status.mirroringStatus.daemonsStatus[]

描述

FilesystemMirrorInfoSpec is the filesystem mirror status of a given filesystem

类型

`object`

属性	类型	描述
`daemon_id`	`integer`	DaemonID is the cephfs-mirror name
`filesystems`	`array`	Filesystems is the list of filesystems managed by a given cephfs-mirror daemon

.status.mirroringStatus.daemonsStatus[].filesystems

描述

Filesystems is the list of filesystems managed by a given cephfs-mirror daemon

类型

`array`

.status.mirroringStatus.daemonsStatus[].filesystems[]

描述

FilesystemsSpec is spec for the mirrored filesystem

类型

`object`

属性	类型	描述
`directory_count`	`integer`	DirectoryCount is the number of directories in the filesystem
`filesystem_id`	`integer`	FilesystemID is the filesystem identifier

属性	类型	描述
`name`	`string`	Name is name of the filesystem
`peers`	`array`	Peers represents the mirroring peers

.status.mirroringStatus.daemonsStatus[].filesystems[].peers

描述

Peers represents the mirroring peers

类型

`array`

.status.mirroringStatus.daemonsStatus[].filesystems[].peers[]

描述

FilesystemMirrorInfoPeerSpec is the specification of a filesystem peer mirror

类型

`object`

属性	类型	描述
`remote`	`object`	Remote are the remote cluster information
`stats`	`object`	Stats are the stat a peer mirror

属性	类型	描述
`uuid`	`string`	UUID is the peer unique identifier

.status.mirroringStatus.daemonsStatus[].filesystems[].peers[].remote

描述

Remote are the remote cluster information

类型

`object`

属性	类型	描述
`client_name`	`string`	ClientName is cephx name
`cluster_name`	`string`	ClusterName is the name of the cluster
`fs_name`	`string`	FsName is the filesystem name

.status.mirroringStatus.daemonsStatus[].filesystems[].peers[].stats

描述

Stats are the stat a peer mirror

类型

`object`

属性	类型	描述
`failure_count`	`integer`	FailureCount is the number of mirroring failure
`recovery_count`	`integer`	RecoveryCount is the number of recovery attempted after failures

.status.snapshotScheduleStatus

描述

FilesystemSnapshotScheduleStatusSpec is the status of the snapshot schedule

类型

`object`

属性	类型	描述
`details`	`string`	Details contains potential status errors
`lastChanged`	`string`	LastChanged is the last time time the status last changed
`lastChecked`	`string`	LastChecked is the last time time the status was checked
`snapshotSchedules`	`array`	SnapshotSchedules is the list of snapshots scheduled

.status.snapshotScheduleStatus.snapshotSchedules

描述

SnapshotSchedules is the list of snapshots scheduled

类型

`array`

.status.snapshotScheduleStatus.snapshotSchedules[]

描述

FilesystemSnapshotSchedulesSpec is the list of snapshot scheduled for images in a pool

类型

`object`

属性	类型	描述
`fs`	`string`	Fs is the name of the Ceph Filesystem
`path`	`string`	Path is the path on the filesystem
`rel_path`	`string`	
`retention`	`object`	FilesystemSnapshotScheduleStatusRetention is the retention specification for a filesystem snapshot schedule
`schedule`	`string`	
`subvol`	`string`	Subvol is the name of the sub volume

.status.snapshotScheduleStatus.snapshotSchedules[].retention

描述

FilesystemSnapshotScheduleStatusRetention is the retention specification for a filesystem snapshot schedule

类型

`object`

属性	类型	描述
`active`	`boolean`	Active is whether the scheduled is active or not
`created`	`string`	Created is when the snapshot schedule was created
`created_count`	`integer`	CreatedCount is total amount of snapshots
`first`	`string`	First is when the first snapshot schedule was taken
`last`	`string`	Last is when the last snapshot schedule was taken
`last_pruned`	`string`	LastPruned is when the last snapshot schedule was pruned
`pruned_count`	`integer`	PrunedCount is total amount of pruned snapshots

属性	类型	描述
`start`	`string`	Start is when the snapshot schedule starts

API 端点

可用的 API 端点如下：

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

/apis/ceph.rook.io/v1/namespaces/{namespace}/cephfilesystems

HTTP 方法

`DELETE`

描述

delete collection of CephFilesystem

HTTP 响应

HTTP 状态码	响应体
200 - OK	`Status` schema
401 - Unauthorized	Empty

HTTP 方法

`GET`

描述

list objects of kind CephFilesystem

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephFilesystemList` schema
401 - Unauthorized	Empty

HTTP 方法

`POST`

描述

create a new CephFilesystem

查询参数

参数	类型	描述
`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

参数	类型	描述
`fieldValidation`	`string`	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.

请求体参数

参数	类型	描述
`body`	`CephFilesystem` schema	`application/json` formatted

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephFilesystem` schema
201 - Created	`CephFilesystem` schema
202 - Accepted	`CephFilesystem` schema
401 - Unauthorized	Empty

/apis/ceph.rook.io/v1/namespaces/{namespace}/cephfilesystems/{name}

HTTP 方法

`DELETE`

描述

delete the specified CephFilesystem

查询参数

参数	类型	描述
`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 响应

HTTP 状态码	响应体
200 - OK	`Status` schema
202 - Accepted	`Status` schema
401 - Unauthorized	Empty

HTTP 方法

`GET`

描述

read the specified CephFilesystem

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephFilesystem` schema
401 - Unauthorized	Empty

HTTP 方法

`PATCH`

描述

partially update the specified CephFilesystem

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	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.

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephFilesystem` schema
401 - Unauthorized	Empty

HTTP 方法

`PUT`

描述

replace the specified CephFilesystem

查询参数

参数	类型	描述
`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

参数	类型	描述
`fieldValidation`	`string`	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.

请求体参数

参数	类型	描述
`body`	`CephFilesystem` schema	`application/json` formatted

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephFilesystem` schema
201 - Created	`CephFilesystem` schema
401 - Unauthorized	Empty

/apis/ceph.rook.io/v1/namespaces/{namespace}/cephfilesystems/{name}/status

HTTP 方法

`GET`

描述

read status of the specified CephFilesystem

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephFilesystem` schema
401 - Unauthorized	Empty

HTTP 方法

`PATCH`

描述

partially update status of the specified CephFilesystem

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	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.

参数	类型	描述
		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 响应

HTTP 状态码	响应体
200 - OK	`CephFilesystem` schema
401 - Unauthorized	Empty

HTTP 方法

`PUT`

描述

replace status of the specified CephFilesystem

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	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,

参数	类型	描述
		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.

请求体参数

参数	类型	描述
`body`	`CephFilesystem` schema	`application/json` formatted

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephFilesystem` schema
201 - Created	`CephFilesystem` schema
401 - Unauthorized	Empty

CephBlockPool [cephblockpools.ceph.rook.io/v1]

描述

CephBlockPool represents a Ceph Storage Pool

类型

`object`

必填

`metadata` `spec`

规格

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

属性	类型	描述
`kind`	`string`	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`metadata`	`ObjectMeta`	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
`spec`	`object`	NamedBlockPoolSpec allows a block pool to be created with a non-default name. This is more specific than the NamedPoolSpec so we get schema validation on the allowed pool names that can be specified.
`status`	`object`	CephBlockPoolStatus represents the mirroring status of Ceph Storage Pool

.spec

描述

NamedBlockPoolSpec allows a block pool to be created with a non-default name. This is more specific than the NamedPoolSpec so we get schema validation on the allowed pool names that can be specified.

类型

`object`

属性	类型	描述
`application`	`string`	The application name to set on the pool. Only expected to be set for rgw pools.
`compressionMode`	`string`	DEPRECATED: use Parameters instead, e.g., Parameters["compression_mode"] = "force" The inline compression mode in Bluestore OSD to set to (options are: none, passive, aggressive, force) Do NOT set a default value for kubebuilder as this will override the Parameters
`crushRoot`	`string`	The root of the crush hierarchy utilized by the pool
`deviceClass`	`string`	The device class the OSD should set to for use in the pool
`enableCrushUpdates`	`boolean`	Allow rook operator to change the pool CRUSH tunables once the pool is created
`enableRBDStats`	`boolean`	EnableRBDStats is used to enable gathering of statistics for all RBD images in the pool
`erasureCoded`	`object`	The erasure code settings
`failureDomain`	`string`	The failure domain: osd/host/(region or zone if available) - technically also any type in the crush

属性	类型	描述
		map
`mirroring`	`object`	The mirroring settings
`name`	`string`	The desired name of the pool if different from the CephBlockPool CR name.
`parameters`	`object`	Parameters is a list of properties to enable on a given pool
`quotas`	`object`	The quota settings
`replicated`	`object`	The replication settings
`statusCheck`	`object`	The mirroring statusCheck

.spec.erasureCoded

描述

The erasure code settings

类型

`object`

必填

`codingChunks` `dataChunks`

属性	类型	描述
`algorithm`	`string`	The algorithm for erasure coding. If absent, defaults to the plugin specified in osd_pool_default_erasure_code_profile.
`codingChunks`	`integer`	Number of coding chunks per object in an erasure coded storage pool (required for erasure-coded pool type). This is the number of OSDs that can be lost simultaneously before data cannot be recovered.
`dataChunks`	`integer`	Number of data chunks per object in an erasure coded storage pool (required for erasure-coded pool type). The number of chunks required to recover an object when any single OSD is lost is the same as dataChunks so be aware that the larger the number of data chunks, the higher the cost of recovery.

.spec.mirroring

描述

The mirroring settings

类型

`object`

属性	类型	描述
`enabled`	`boolean`	Enabled whether this pool is mirrored or not

属性	类型	描述
`mode`	`string`	Mode is the mirroring mode: pool, image or init-only.
`peers`	`object`	Peers represents the peers spec
`snapshotSchedules`	`array`	SnapshotSchedules is the scheduling of snapshot for mirrored images/pools

.spec.mirroring.peers

描述

Peers represents the peers spec

类型

`object`

属性	类型	描述
`secretNames`	`array`	SecretNames represents the Kubernetes Secret names to add rbd-mirror or cephfs-mirror peers

.spec.mirroring.peers.secretNames

描述

SecretNames represents the Kubernetes Secret names to add rbd-mirror or cephfs-mirror peers

类型

`array`

.spec.mirroring.peers.secretNames[]

类型

`string`

.spec.mirroring.snapshotSchedules

描述

SnapshotSchedules is the scheduling of snapshot for mirrored images/pools

类型

`array`

.spec.mirroring.snapshotSchedules[]

描述

SnapshotScheduleSpec represents the snapshot scheduling settings of a mirrored pool

类型

`object`

属性	类型	描述
`interval`	`string`	Interval represent the periodicity of the snapshot.
`path`	`string`	Path is the path to snapshot, only valid for CephFS
`startTime`	`string`	StartTime indicates when to start the snapshot

.spec.parameters

描述

Parameters is a list of properties to enable on a given pool

类型

`object`

.spec.quotas

描述

The quota settings

类型

`object`

属性	类型	描述
`maxBytes`	`integer`	MaxBytes represents the quota in bytes Deprecated in favor of MaxSize
`maxObjects`	`integer`	MaxObjects represents the quota in objects
`maxSize`	`string`	MaxSize represents the quota in bytes as a string

.spec.replicated

描述

The replication settings

类型

`object`

必填

`size`

属性	类型	描述
`hybridStorage`	`object`	HybridStorage represents hybrid storage tier settings
`replicasPerFailureDomain`	`integer`	ReplicasPerFailureDomain the number of replica in the specified failure domain
`requireSafeReplicaSize`	`boolean`	RequireSafeReplicaSize if false allows you to set replica 1
`size`	`integer`	Size - Number of copies per object in a replicated storage pool, including the object itself (required for replicated pool type)
`subFailureDomain`	`string`	SubFailureDomain the name of the sub-failure domain
`targetSizeRatio`	`number`	TargetSizeRatio gives a hint (%) to Ceph in terms of expected consumption of the total cluster capacity

.spec.replicated.hybridStorage

描述

HybridStorage represents hybrid storage tier settings

类型

`object`

必填

`primaryDeviceClass` `secondaryDeviceClass`

属性	类型	描述
`primaryDeviceClass`	`string`	PrimaryDeviceClass represents high performance tier (for example SSD or NVME) for Primary OSD
`secondaryDeviceClass`	`string`	SecondaryDeviceClass represents low performance tier (for example HDDs) for remaining OSDs

.spec.statusCheck

描述

The mirroring statusCheck

类型

`object`

属性	类型	描述
`mirror`	`object`	HealthCheckSpec represents the health check of an object store bucket

.spec.statusCheck.mirror

描述

HealthCheckSpec represents the health check of an object store bucket

类型

`object`

属性	类型	描述
`disabled`	`boolean`	
`interval`	`string`	Interval is the internal in second or minute for the health check to run like 60s for 60 seconds
`timeout`	`string`	

.status

描述

CephBlockPoolStatus represents the mirroring status of Ceph Storage Pool

类型

`object`

属性	类型	描述
`cephx`	`object`	PeerTokenCephxStatus represents the cephx key rotation status for peer tokens
`conditions`	`array`	
`info`	`object`	
`mirroringInfo`	`object`	MirroringInfoSpec is the status of the pool/radosnamespace mirroring

属性	类型	描述
`mirroringStatus`	`object`	MirroringStatusSpec is the status of the pool/radosNamespace mirroring
`observedGeneration`	`integer`	ObservedGeneration is the latest generation observed by the controller.
`phase`	`string`	ConditionType represent a resource's status
`poolID`	`integer`	optional
`snapshotScheduleStatus`	`object`	SnapshotScheduleStatusSpec is the status of the snapshot schedule

.status.cephx

描述

PeerTokenCephxStatus represents the cephx key rotation status for peer tokens

类型

`object`

属性	类型	描述
`peerToken`	`object`	PeerToken shows the rotation status of the peer token associated with the `rbd-mirror-peer` user.

.status.cephx.peerToken

描述

PeerToken shows the rotation status of the peer token associated with the `rbd-mirror-peer` user.

类型

`object`

属性	类型	描述
`keyCephVersion`	`string`	KeyCephVersion reports the Ceph version that created the current generation's keys. This is same string format as reported by `CephCluster.status.version.version` to allow them to be compared. E.g., `20.2.0-0`. For all newly-created resources, this field set to the version of Ceph that created the key. The special value "Uninitialized" indicates that keys are being created for the first time. An empty string indicates that the version is unknown, as expected in brownfield deployments.
`keyGeneration`	`integer`	KeyGeneration represents the CephX key generation for the last successful reconcile. For all newly-created resources, this field is set to `1`. When keys are rotated due to any rotation policy, the generation is incremented or updated to the configured policy generation. Generation `0` indicates that keys existed prior to the implementation of key tracking.

.status.conditions

类型

`array`

.status.conditions[]

描述

Condition represents a status condition on any Rook-Ceph Custom Resource.

类型

`object`

属性	类型	描述
`lastHeartbeatTime`	`string`	
`lastTransitionTime`	`string`	
`message`	`string`	
`reason`	`string`	ConditionReason is a reason for a condition
`status`	`string`	
`type`	`string`	ConditionType represent a resource's status

.status.info

类型

`object`

.status.mirroringInfo

描述

MirroringInfoSpec is the status of the pool/radosnamespace mirroring

类型

`object`

属性	类型	描述
`details`	`string`	
`lastChanged`	`string`	
`lastChecked`	`string`	
`mode`	`string`	Mode is the mirroring mode
`peers`	`array`	Peers are the list of peer sites connected to that cluster
`site_name`	`string`	SiteName is the current site name

.status.mirroringInfo.peers

描述

Peers are the list of peer sites connected to that cluster

类型

`array`

.status.mirroringInfo.peers[]

描述

PeersSpec contains peer details

类型

`object`

属性	类型	描述
`client_name`	`string`	ClientName is the CephX user used to connect to the peer
`direction`	`string`	Direction is the peer mirroring direction
`mirror_uuid`	`string`	MirrorUUID is the mirror UUID
`site_name`	`string`	SiteName is the current site name
`uuid`	`string`	UUID is the peer UUID

.status.mirroringStatus

描述

MirroringStatusSpec is the status of the pool/radosNamespace mirroring

类型

`object`

属性	类型	描述
`details`	`string`	Details contains potential status errors
`lastChanged`	`string`	LastChanged is the last time time the status last changed

属性	类型	描述
`lastChecked`	`string`	LastChecked is the last time time the status was checked
`summary`	`object`	Summary is the mirroring status summary

.status.mirroringStatus.summary

描述

Summary is the mirroring status summary

类型

`object`

属性	类型	描述
`daemon_health`	`string`	DaemonHealth is the health of the mirroring daemon
`group_health`	`string`	GroupHealth is the health of the mirrored image group
`group_states`	`object`	GroupStates is the various state for all mirrored image groups
`health`	`string`	Health is the mirroring health
`image_health`	`string`	ImageHealth is the health of the mirrored image

属性	类型	描述
`image_states`	`object`	ImageStates is the various state for all mirrored images
`states`	`object`	States is the various state for all mirrored images

.status.mirroringStatus.summary.group_states

描述

GroupStates is the various state for all mirrored image groups

类型

`object`

属性	类型	描述
`error`	`integer`	Error is when the mirroring state is errored
`replaying`	`integer`	Replaying is when the replay of the mirroring journal is on-going
`starting_replay`	`integer`	StartingReplay is when the replay of the mirroring journal starts
`stopped`	`integer`	Stopped is when the mirroring state is stopped

属性	类型	描述
`stopping_replay`	`integer`	StopReplaying is when the replay of the mirroring journal stops
`syncing`	`integer`	Syncing is when the image is syncing
`unknown`	`integer`	Unknown is when the mirroring state is unknown

.status.mirroringStatus.summary.image_states

描述

ImageStates is the various state for all mirrored images

类型

`object`

属性	类型	描述
`error`	`integer`	Error is when the mirroring state is errored
`replaying`	`integer`	Replaying is when the replay of the mirroring journal is on-going
`starting_replay`	`integer`	StartingReplay is when the replay of the mirroring journal starts

属性	类型	描述
`stopped`	`integer`	Stopped is when the mirroring state is stopped
`stopping_replay`	`integer`	StopReplaying is when the replay of the mirroring journal stops
`syncing`	`integer`	Syncing is when the image is syncing
`unknown`	`integer`	Unknown is when the mirroring state is unknown

.status.mirroringStatus.summary.states

描述

States is the various state for all mirrored images

类型

`object`

属性	类型	描述
`error`	`integer`	Error is when the mirroring state is errored
`replaying`	`integer`	Replaying is when the replay of the mirroring journal is on-going

属性	类型	描述
`starting_replay`	`integer`	StartingReplay is when the replay of the mirroring journal starts
`stopped`	`integer`	Stopped is when the mirroring state is stopped
`stopping_replay`	`integer`	StopReplaying is when the replay of the mirroring journal stops
`syncing`	`integer`	Syncing is when the image is syncing
`unknown`	`integer`	Unknown is when the mirroring state is unknown

.status.snapshotScheduleStatus

描述

SnapshotScheduleStatusSpec is the status of the snapshot schedule

类型

`object`

属性	类型	描述
`details`	`string`	Details contains potential status errors

属性	类型	描述
`lastChanged`	`string`	LastChanged is the last time time the status last changed
`lastChecked`	`string`	LastChecked is the last time time the status was checked
`snapshotSchedules`	`array`	SnapshotSchedules is the list of snapshots scheduled

.status.snapshotScheduleStatus.snapshotSchedules

描述

SnapshotSchedules is the list of snapshots scheduled

类型

`array`

.status.snapshotScheduleStatus.snapshotSchedules[]

描述

SnapshotSchedulesSpec is the list of snapshot scheduled for images in a pool

类型

`object`

属性	类型	描述
`image`	`string`	Image is the mirrored image

属性	类型	描述
`items`	`array`	Items is the list schedules times for a given snapshot
`namespace`	`string`	Namespace is the RADOS namespace the image is part of
`pool`	`string`	Pool is the pool name

.status.snapshotScheduleStatus.snapshotSchedules[].items

描述

Items is the list schedules times for a given snapshot

类型

`array`

.status.snapshotScheduleStatus.snapshotSchedules[].items[]

描述

SnapshotSchedule is a schedule

类型

`object`

属性	类型	描述
`interval`	`string`	Interval is the interval in which snapshots will be taken

属性	类型	描述
`start_time`	`string`	StartTime is the snapshot starting time

API 端点

可用的 API 端点如下：

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

/apis/ceph.rook.io/v1/namespaces/{namespace}/cephblockpools

HTTP 方法

`DELETE`

描述

delete collection of CephBlockPool

HTTP 响应

HTTP 状态码	响应体
200 - OK	`Status` schema
401 - Unauthorized	Empty

HTTP 方法

`GET`

描述

list objects of kind CephBlockPool

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephBlockPoolList` schema
401 - Unauthorized	Empty

HTTP 方法

`POST`

描述

create a new CephBlockPool

查询参数

参数	类型	描述
`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

参数	类型	描述
`fieldValidation`	`string`	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.

请求体参数

参数	类型	描述
`body`	`CephBlockPool` schema	`application/json` formatted

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephBlockPool` schema
201 - Created	`CephBlockPool` schema
202 - Accepted	`CephBlockPool` schema
401 - Unauthorized	Empty

/apis/ceph.rook.io/v1/namespaces/{namespace}/cephblockpools/{name}

HTTP 方法

`DELETE`

描述

delete the specified CephBlockPool

查询参数

参数	类型	描述
`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 响应

HTTP 状态码	响应体
200 - OK	`Status` schema
202 - Accepted	`Status` schema
401 - Unauthorized	Empty

HTTP 方法

`GET`

描述

read the specified CephBlockPool

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephBlockPool` schema
401 - Unauthorized	Empty

HTTP 方法

`PATCH`

描述

partially update the specified CephBlockPool

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	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.

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephBlockPool` schema
401 - Unauthorized	Empty

HTTP 方法

`PUT`

描述

replace the specified CephBlockPool

查询参数

参数	类型	描述
`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

参数	类型	描述
`fieldValidation`	`string`	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.

请求体参数

参数	类型	描述
`body`	`CephBlockPool` schema	`application/json` formatted

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephBlockPool` schema
201 - Created	`CephBlockPool` schema
401 - Unauthorized	Empty

/apis/ceph.rook.io/v1/namespaces/{namespace}/cephblockpools/{name}/status

HTTP 方法

`GET`

描述

read status of the specified CephBlockPool

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephBlockPool` schema
401 - Unauthorized	Empty

HTTP 方法

`PATCH`

描述

partially update status of the specified CephBlockPool

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	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.

参数	类型	描述
		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 响应

HTTP 状态码	响应体
200 - OK	`CephBlockPool` schema
401 - Unauthorized	Empty

HTTP 方法

`PUT`

描述

replace status of the specified CephBlockPool

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	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,

参数	类型	描述
		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.

请求体参数

参数	类型	描述
`body`	`CephBlockPool` schema	`application/json` formatted

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephBlockPool` schema
201 - Created	`CephBlockPool` schema
401 - Unauthorized	Empty

CephObjectStore [cephobjectstores.ceph.rook.io/v1]

描述

CephObjectStore represents a Ceph Object Store Gateway

类型

`object`

必填

`metadata` `spec`

规格

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

属性	类型	描述
`kind`	`string`	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`metadata`	`ObjectMeta`	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
`spec`	`object`	ObjectStoreSpec represent the spec of a pool
`status`	`object`	ObjectStoreStatus represents the status of a Ceph Object Store resource

.spec

描述

ObjectStoreSpec represent the spec of a pool

类型

`object`

属性	类型	描述
`allowUsersInNamespaces`	`array`	The list of allowed namespaces in addition to the object store namespace where ceph object store users may be created. Specify "*"

属性	类型	描述
		to allow all namespaces, otherwise list individual namespaces that are to be allowed. This is useful for applications that need object store credentials to be created in their own namespace, where neither OBCs nor COSI is being used to create buckets. The default is empty.
`auth`	`object`	The authentication configuration
`dataPool`	`object`	The data pool settings
`defaultRealm`	`boolean`	Set this realm as the default in Ceph. Only one realm should be default. Do not set this true on more than one CephObjectStore. This may not be set when zone is also specified; in this case, the realm referenced by the zone's zonegroup should configure defaulting behavior.
`gateway`	`object`	The rgw pod info
`healthCheck`	`object`	The RGW health probes
`hosting`	`object`	Hosting settings for the object store. A common use case for hosting configuration is to inform Rook of endpoints that support DNS

属性	类型	描述
		wildcards, which in turn allows virtual host-style bucket addressing.
`metadataPool`	`object`	The metadata pool settings
`preservePoolsOnDelete`	`boolean`	Preserve pools on object store deletion
`protocols`	`object`	The protocol specification
`security`	`object`	Security represents security settings
`sharedPools`	`object`	The pool information when configuring RADOS namespaces in existing pools.
`zone`	`object`	The multisite info

.spec.allowUsersInNamespaces

描述

The list of allowed namespaces in addition to the object store namespace where ceph object store users may be created. Specify "*" to allow all namespaces, otherwise list individual namespaces that are to be allowed. This is useful for applications that need object store credentials to be created in their own namespace, where neither OBCs nor COSI is being used to create buckets. The default is empty.

类型

`array`

.spec.allowUsersInNamespaces[]

类型

`string`

.spec.auth

描述

The authentication configuration

类型

`object`

属性	类型	描述
`keystone`	`object`	The spec for Keystone

.spec.auth.keystone

描述

The spec for Keystone

类型

`object`

必填

`acceptedRoles` `serviceUserSecretName` `url`

属性	类型	描述
`acceptedRoles`	`array`	The roles requires to serve requests.
`implicitTenants`	`string`	Create new users in their own tenants of the same name. Possible values are true, false, swift and s3. The latter have the effect of splitting the identity space such that only the indicated protocol will use implicit tenants.
`revocationInterval`	`integer`	The number of seconds between token revocation checks.
`serviceUserSecretName`	`string`	The name of the secret containing the credentials for the service user account used by RGW. It has to be in the same namespace as the object store resource.
`tokenCacheSize`	`integer`	The maximum number of entries in each Keystone token cache.
`url`	`string`	The URL for the Keystone server.

.spec.auth.keystone.acceptedRoles

描述

The roles requires to serve requests.

类型

`array`

.spec.auth.keystone.acceptedRoles[]

类型

`string`

.spec.dataPool

描述

The data pool settings

类型

`object`

属性	类型	描述
`application`	`string`	The application name to set on the pool. Only expected to be set for rgw pools.
`compressionMode`	`string`	DEPRECATED: use Parameters instead, e.g., Parameters["compression_mode"] = "force" The inline compression mode in Bluestore OSD to set to (options are: none, passive, aggressive, force) Do NOT set a default value for kubebuilder as this will override the Parameters
`crushRoot`	`string`	The root of the crush hierarchy utilized by the pool

属性	类型	描述
`deviceClass`	`string`	The device class the OSD should set to for use in the pool
`enableCrushUpdates`	`boolean`	Allow rook operator to change the pool CRUSH tunables once the pool is created
`enableRBDStats`	`boolean`	EnableRBDStats is used to enable gathering of statistics for all RBD images in the pool
`erasureCoded`	`object`	The erasure code settings
`failureDomain`	`string`	The failure domain: osd/host/(region or zone if available) - technically also any type in the crush map
`mirroring`	`object`	The mirroring settings
`parameters`	`object`	Parameters is a list of properties to enable on a given pool
`quotas`	`object`	The quota settings

属性	类型	描述
`replicated`	`object`	The replication settings
`statusCheck`	`object`	The mirroring statusCheck

.spec.dataPool.erasureCoded

描述

The erasure code settings

类型

`object`

必填

`codingChunks` `dataChunks`

属性	类型	描述
`algorithm`	`string`	The algorithm for erasure coding. If absent, defaults to the plugin specified in osd_pool_default_erasure_code_profile.
`codingChunks`	`integer`	Number of coding chunks per object in an erasure coded storage pool (required for erasure-coded pool type). This is the number of OSDs that can be lost simultaneously before data cannot be recovered.
`dataChunks`	`integer`	Number of data chunks per object in an erasure coded storage pool (required for erasure-coded pool type). The

属性	类型	描述
		number of chunks required to recover an object when any single OSD is lost is the same as dataChunks so be aware that the larger the number of data chunks, the higher the cost of recovery.

.spec.dataPool.mirroring

描述

The mirroring settings

类型

`object`

属性	类型	描述
`enabled`	`boolean`	Enabled whether this pool is mirrored or not
`mode`	`string`	Mode is the mirroring mode: pool, image or init-only.
`peers`	`object`	Peers represents the peers spec
`snapshotSchedules`	`array`	SnapshotSchedules is the scheduling of snapshot for mirrored images/pools

.spec.dataPool.mirroring.peers

描述

Peers represents the peers spec

类型

`object`

属性	类型	描述
`secretNames`	`array`	SecretNames represents the Kubernetes Secret names to add rbd-mirror or cephfs-mirror peers

.spec.dataPool.mirroring.peers.secretNames

描述

SecretNames represents the Kubernetes Secret names to add rbd-mirror or cephfs-mirror peers

类型

`array`

.spec.dataPool.mirroring.peers.secretNames[]

类型

`string`

.spec.dataPool.mirroring.snapshotSchedules

描述

SnapshotSchedules is the scheduling of snapshot for mirrored images/pools

类型

`array`

.spec.dataPool.mirroring.snapshotSchedules[]

描述

SnapshotScheduleSpec represents the snapshot scheduling settings of a mirrored pool

类型

`object`

属性	类型	描述
`interval`	`string`	Interval represent the periodicity of the snapshot.
`path`	`string`	Path is the path to snapshot, only valid for CephFS
`startTime`	`string`	StartTime indicates when to start the snapshot

.spec.dataPool.parameters

描述

Parameters is a list of properties to enable on a given pool

类型

`object`

.spec.dataPool.quotas

描述

The quota settings

类型

`object`

属性	类型	描述
`maxBytes`	`integer`	MaxBytes represents the quota in bytes Deprecated in favor of MaxSize
`maxObjects`	`integer`	MaxObjects represents the quota in objects
`maxSize`	`string`	MaxSize represents the quota in bytes as a string

.spec.dataPool.replicated

描述

The replication settings

类型

`object`

必填

`size`

属性	类型	描述
`hybridStorage`	`object`	HybridStorage represents hybrid storage tier settings
`replicasPerFailureDomain`	`integer`	ReplicasPerFailureDomain the number of replica in the specified failure domain

属性	类型	描述
`requireSafeReplicaSize`	`boolean`	RequireSafeReplicaSize if false allows you to set replica 1
`size`	`integer`	Size - Number of copies per object in a replicated storage pool, including the object itself (required for replicated pool type)
`subFailureDomain`	`string`	SubFailureDomain the name of the sub-failure domain
`targetSizeRatio`	`number`	TargetSizeRatio gives a hint (%) to Ceph in terms of expected consumption of the total cluster capacity

.spec.dataPool.replicated.hybridStorage

描述

HybridStorage represents hybrid storage tier settings

类型

`object`

必填

`primaryDeviceClass` `secondaryDeviceClass`

属性	类型	描述
`primaryDeviceClass`	`string`	PrimaryDeviceClass represents high performance tier (for example SSD or NVME) for Primary OSD
`secondaryDeviceClass`	`string`	SecondaryDeviceClass represents low performance tier (for example HDDs) for remaining OSDs

.spec.dataPool.statusCheck

描述

The mirroring statusCheck

类型

`object`

属性	类型	描述
`mirror`	`object`	HealthCheckSpec represents the health check of an object store bucket

.spec.dataPool.statusCheck.mirror

描述

HealthCheckSpec represents the health check of an object store bucket

类型

`object`

属性	类型	描述
`disabled`	`boolean`	
`interval`	`string`	Interval is the internal in second or minute for the health check to run like 60s for 60 seconds
`timeout`	`string`	

.spec.gateway

描述

The rgw pod info

类型

`object`

属性	类型	描述
`additionalVolumeMounts`	`array`	AdditionalVolumeMounts allows additional volumes to be mounted to the RGW pod. The root directory for each additional volume mount is `/var/rgw`. Example: for an additional mount at subPath `ldap`, mounted from a secret that has key `bindpass.secret`, the file would reside at `/var/rgw/ldap/bindpass.secret`.
`annotations`	`object`	The annotations-related configuration to add/set on each Pod related object.

属性	类型	描述
`caBundleRef`	`string`	The name of the secret that stores custom ca-bundle with root and intermediate certificates.
`dashboardEnabled`	`boolean`	Whether rgw dashboard is enabled for the rgw daemon. If not set, the rgw dashboard will be enabled.
`disableMultisiteSyncTraffic`	`boolean`	DisableMultisiteSyncTraffic, when true, prevents this object store's gateways from transmitting multisite replication data. Note that this value does not affect whether gateways receive multisite replication traffic: see ObjectZone.spec.customEndpoints for that. If false or unset, this object store's gateways will be able to transmit multisite replication data.
`externalRgwEndpoints`	`array`	ExternalRgwEndpoints points to external RGW endpoint(s). Multiple endpoints can be given, but for stability of ObjectBucketClaims, we highly recommend that users give only a single external RGW endpoint that is a load balancer that sends requests to the multiple RGWs.

属性	类型	描述
`hostNetwork`	`boolean`	Whether host networking is enabled for the rgw daemon. If not set, the network settings from the cluster CR will be applied.
`instances`	`integer`	The number of pods in the rgw replicaset.
`labels`	`object`	The labels-related configuration to add/set on each Pod related object.
`opsLogSidecar`	`object`	Enable enhanced operation Logs for S3 in a sidecar named ops-log
`placement`	`object`	
`port`	`integer`	The port the rgw service will be listening on (http)
`priorityClassName`	`string`	PriorityClassName sets priority classes on the rgw pods

属性	类型	描述
`readAffinity`	`object`	ReadAffinity defines the RGW read affinity policy to optimize the read requests for the RGW clients Note: Only supported from Ceph Tentacle (v20)
`resources`	`object`	The resource requirements for the rgw pods
`rgwCommandFlags`	`object`	RgwCommandFlags sets Ceph RGW config values for the gateway clients that serve this object store. Values are modified at RGW startup, resulting in RGW pod restarts. This feature is intended for advanced users. It allows breaking configurations to be easily applied. Use with caution.
`rgwConfig`	`object`	RgwConfig sets Ceph RGW config values for the gateway clients that serve this object store. Values are modified at runtime without RGW restart. This feature is intended for advanced users. It allows breaking configurations to be easily applied. Use with caution.
`rgwConfigFromSecret`	`object`	RgwConfigFromSecret works exactly like RgwConfig but takes config value from Secret Key reference. Values are modified at runtime without RGW

属性	类型	描述
		restart. This feature is intended for advanced users. It allows breaking configurations to be easily applied. Use with caution.
`securePort`	`integer`	The port the rgw service will be listening on (https)
`service`	`object`	The configuration related to add/set on each rgw service.
`sslCertificateRef`	`string`	The name of the secret that stores the ssl certificate for secure rgw connections

.spec.gateway.additionalVolumeMounts

描述

AdditionalVolumeMounts allows additional volumes to be mounted to the RGW pod. The root directory for each additional volume mount is `/var/rgw`. Example: for an additional mount at subPath `ldap`, mounted from a secret that has key `bindpass.secret`, the file would reside at `/var/rgw/ldap/bindpass.secret`.

类型

`array`

.spec.gateway.additionalVolumeMounts[]

描述

AdditionalVolumeMount represents the source from where additional files in pod containers should come from and what subdirectory they are made available in.

类型

`object`

必填

`subPath` `volumeSource`

属性	类型	描述
`subPath`	`string`	SubPath defines the sub-path (subdirectory) of the directory root where the volumeSource will be mounted. All files/keys in the volume source's volume will be mounted to the subdirectory. This is not the same as the Kubernetes `subPath` volume mount option. Each subPath definition must be unique and must not contain ':'.
`volumeSource`	`object`	

.spec.gateway.additionalVolumeMounts[].volumeSource

类型

`object`

属性	类型	描述
`configMap`	`object`	
`emptyDir`	`object`	
`hostPath`	`object`	
`persistentVolumeClaim`	`object`	

属性	类型	描述
`projected`	`object`	
`secret`	`object`	

.spec.gateway.additionalVolumeMounts[].volumeSource.configMap

类型

`object`

属性	类型	描述
`defaultMode`	`integer`	
`items`	`array`	
`name`	`string`	
`optional`	`boolean`	

.spec.gateway.additionalVolumeMounts[].volumeSource.configMap.items

类型

`array`

.spec.gateway.additionalVolumeMounts[].volumeSource.configMap.items[]

类型

`object`

必填

`key` `path`

属性	类型	描述
`key`	`string`	
`mode`	`integer`	
`path`	`string`	

.spec.gateway.additionalVolumeMounts[].volumeSource.emptyDir

类型

`object`

属性	类型	描述
`medium`	`string`	
`sizeLimit`	`integer|string`	

.spec.gateway.additionalVolumeMounts[].volumeSource.hostPath

类型

`object`

必填

`path`

属性	类型	描述
`path`	`string`	

属性	类型	描述
`type`	`string`	

.spec.gateway.additionalVolumeMounts[].volumeSource.persistentVolumeClaim

类型

`object`

必填

`claimName`

属性	类型	描述
`claimName`	`string`	
`readOnly`	`boolean`	

.spec.gateway.additionalVolumeMounts[].volumeSource.projected

类型

`object`

属性	类型	描述
`defaultMode`	`integer`	
`sources`	`array`	

.spec.gateway.additionalVolumeMounts[].volumeSource.projected.sources

类型

`array`

.spec.gateway.additionalVolumeMounts[].volumeSource.projected.sources[]

类型

`object`

属性	类型	描述
`clusterTrustBundle`	`object`	
`configMap`	`object`	
`downwardAPI`	`object`	
`podCertificate`	`object`	
`secret`	`object`	
`serviceAccountToken`	`object`	

.spec.gateway.additionalVolumeMounts[].volumeSource.projected.sources[].clusterTrustBundle

类型

`object`

必填

`path`

属性	类型	描述
`labelSelector`	`object`	

属性	类型	描述
`name`	`string`	
`optional`	`boolean`	
`path`	`string`	
`signerName`	`string`	

.spec.gateway.additionalVolumeMounts[].volumeSource.projected.sources[].clusterTrustBundle.labelSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.gateway.additionalVolumeMounts[].volumeSource.projected.sources[].clusterTrustBundle.labelSelector.matchExpressions

类型

`array`

.spec.gateway.additionalVolumeMounts[].volumeSource.projected.sources[].clusterTrustBundle.labelSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.gateway.additionalVolumeMounts[].volumeSource.projected.sources[].clusterTrustBundle.labelSelector.matchExpressions[].values

类型

`array`

.spec.gateway.additionalVolumeMounts[].volumeSource.projected.sources[].clusterTrustBundle.labelSelector.matchExpressions[].values[]

类型

`string`

.spec.gateway.additionalVolumeMounts[].volumeSource.projected.sources[].clusterTrustBundle.labelSelector.matchLabels

类型

`object`

.spec.gateway.additionalVolumeMounts[].volumeSource.projected.sources[].configMap

类型

`object`

属性	类型	描述
`items`	`array`	
`name`	`string`	
`optional`	`boolean`	

.spec.gateway.additionalVolumeMounts[].volumeSource.projected.sources[].configMap.items

类型

`array`

.spec.gateway.additionalVolumeMounts[].volumeSource.projected.sources[].configMap.items[]

类型

`object`

必填

`key` `path`

属性	类型	描述
`key`	`string`	

属性	类型	描述
`mode`	`integer`	
`path`	`string`	

.spec.gateway.additionalVolumeMounts[].volumeSource.projected.sources[].downwardAPI

类型

`object`

属性	类型	描述
`items`	`array`	

.spec.gateway.additionalVolumeMounts[].volumeSource.projected.sources[].downwardAPI.items

类型

`array`

.spec.gateway.additionalVolumeMounts[].volumeSource.projected.sources[].downwardAPI.items[]

类型

`object`

必填

`path`

属性	类型	描述
`fieldRef`	`object`	
`mode`	`integer`	
`path`	`string`	
`resourceFieldRef`	`object`	

.spec.gateway.additionalVolumeMounts[].volumeSource.projected.sources[].downwardAPI.items[].fieldRef

类型

`object`

必填

`fieldPath`

属性	类型	描述
`apiVersion`	`string`	
`fieldPath`	`string`	

.spec.gateway.additionalVolumeMounts[].volumeSource.projected.sources[].downwardAPI.items[].resourceFieldRef

类型

`object`

必填

`resource`

属性	类型	描述
`containerName`	`string`	
`divisor`	`integer|string`	
`resource`	`string`	

.spec.gateway.additionalVolumeMounts[].volumeSource.projected.sources[].podCertificate

类型

`object`

必填

`keyType` `signerName`

属性	类型	描述
`certificateChainPath`	`string`	
`credentialBundlePath`	`string`	
`keyPath`	`string`	
`keyType`	`string`	
`maxExpirationSeconds`	`integer`	
`signerName`	`string`	
`userAnnotations`	`object`	

.spec.gateway.additionalVolumeMounts[].volumeSource.projected.sources[].podCertificate.userAnnotations

类型

`object`

.spec.gateway.additionalVolumeMounts[].volumeSource.projected.sources[].secret

类型

`object`

属性	类型	描述
`items`	`array`	
`name`	`string`	
`optional`	`boolean`	

.spec.gateway.additionalVolumeMounts[].volumeSource.projected.sources[].secret.items

类型

`array`

.spec.gateway.additionalVolumeMounts[].volumeSource.projected.sources[].secret.items[]

类型

`object`

必填

`key` `path`

属性	类型	描述
`key`	`string`	

属性	类型	描述
`mode`	`integer`	
`path`	`string`	

.spec.gateway.additionalVolumeMounts[].volumeSource.projected.sources[].serviceAccountToken

类型

`object`

必填

`path`

属性	类型	描述
`audience`	`string`	
`expirationSeconds`	`integer`	
`path`	`string`	

.spec.gateway.additionalVolumeMounts[].volumeSource.secret

类型

`object`

属性	类型	描述
`defaultMode`	`integer`	
`items`	`array`	

属性	类型	描述
`optional`	`boolean`	
`secretName`	`string`	

.spec.gateway.additionalVolumeMounts[].volumeSource.secret.items

类型

`array`

.spec.gateway.additionalVolumeMounts[].volumeSource.secret.items[]

类型

`object`

必填

`key` `path`

属性	类型	描述
`key`	`string`	
`mode`	`integer`	
`path`	`string`	

.spec.gateway.annotations

描述

The annotations-related configuration to add/set on each Pod related object.

类型

`object`

.spec.gateway.externalRgwEndpoints

描述

ExternalRgwEndpoints points to external RGW endpoint(s). Multiple endpoints can be given, but for stability of ObjectBucketClaims, we highly recommend that users give only a single external RGW endpoint that is a load balancer that sends requests to the multiple RGWs.

类型

`array`

.spec.gateway.externalRgwEndpoints[]

描述

EndpointAddress is a tuple that describes a single IP address or host name. This is a subset of Kubernetes's v1.EndpointAddress.

类型

`object`

属性	类型	描述
`hostname`	`string`	The DNS-addressable Hostname of this endpoint. This field will be preferred over IP if both are given.
`ip`	`string`	The IP of this endpoint. As a legacy behavior, this supports being given a DNS-addressable hostname as well.

.spec.gateway.labels

描述

The labels-related configuration to add/set on each Pod related object.

类型

`object`

.spec.gateway.opsLogSidecar

描述

Enable enhanced operation Logs for S3 in a sidecar named ops-log

类型

`object`

属性	类型	描述
`resources`	`object`	Resources represents the way to specify resource requirements for the ops-log sidecar

.spec.gateway.opsLogSidecar.resources

描述

Resources represents the way to specify resource requirements for the ops-log sidecar

类型

`object`

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

.spec.gateway.opsLogSidecar.resources.claims

描述

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

类型

`array`

.spec.gateway.opsLogSidecar.resources.claims[]

描述

ResourceClaim references one entry in PodSpec.ResourceClaims.

类型

`object`

必填

`name`

属性	类型	描述
`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.gateway.opsLogSidecar.resources.limits

描述

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

类型

`object`

.spec.gateway.opsLogSidecar.resources.requests

描述

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/

类型

`object`

.spec.gateway.placement

类型

`object`

属性	类型	描述
`nodeAffinity`	`object`	
`podAffinity`	`object`	
`podAntiAffinity`	`object`	
`tolerations`	`array`	
`topologySpreadConstraints`	`array`	

.spec.gateway.placement.nodeAffinity

类型

`object`

属性	类型	描述
`preferredDuringSchedulingIgnoredDuringExecution`	`array`	
`requiredDuringSchedulingIgnoredDuringExecution`	`object`	

.spec.gateway.placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution

类型

`array`

.spec.gateway.placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

类型

`object`

必填

`preference` `weight`

属性	类型	描述
`preference`	`object`	
`weight`	`integer`	

.spec.gateway.placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchFields`	`array`	

.spec.gateway.placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions

类型

`array`

.spec.gateway.placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.gateway.placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values

类型

`array`

.spec.gateway.placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values[]

类型

`string`

.spec.gateway.placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields

类型

`array`

.spec.gateway.placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.gateway.placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values

类型

`array`

.spec.gateway.placement.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values[]

类型

`string`

.spec.gateway.placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution

类型

`object`

必填

`nodeSelectorTerms`

属性	类型	描述
`nodeSelectorTerms`	`array`	

.spec.gateway.placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms

类型

`array`

.spec.gateway.placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[]

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchFields`	`array`	

.spec.gateway.placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions

类型

`array`

.spec.gateway.placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	

属性	类型	描述
`operator`	`string`	
`values`	`array`	

.spec.gateway.placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values

类型

`array`

.spec.gateway.placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values[]

类型

`string`

.spec.gateway.placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields

类型

`array`

.spec.gateway.placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.gateway.placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values

类型

`array`

.spec.gateway.placement.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values[]

类型

`string`

.spec.gateway.placement.podAffinity

类型

`object`

属性	类型	描述
`preferredDuringSchedulingIgnoredDuringExecution`	`array`	
`requiredDuringSchedulingIgnoredDuringExecution`	`array`	

.spec.gateway.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution

类型

`array`

.spec.gateway.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

类型

`object`

必填

`podAffinityTerm` `weight`

属性	类型	描述
`podAffinityTerm`	`object`	
`weight`	`integer`	

.spec.gateway.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm

类型

`object`

必填

`topologyKey`

属性	类型	描述
`labelSelector`	`object`	
`matchLabelKeys`	`array`	
`mismatchLabelKeys`	`array`	
`namespaceSelector`	`object`	
`namespaces`	`array`	
`topologyKey`	`string`	

.spec.gateway.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.gateway.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions

类型

`array`

.spec.gateway.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.gateway.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

类型

`array`

.spec.gateway.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]

类型

`string`

.spec.gateway.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels

类型

`object`

.spec.gateway.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys

类型

`array`

.spec.gateway.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]

类型

`string`

.spec.gateway.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys

类型

`array`

.spec.gateway.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]

类型

`string`

.spec.gateway.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.gateway.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions

类型

`array`

.spec.gateway.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.gateway.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values

类型

`array`

.spec.gateway.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]

类型

`string`

.spec.gateway.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

类型

`object`

.spec.gateway.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

类型

`array`

.spec.gateway.placement.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

类型

`string`

.spec.gateway.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution

类型

`array`

.spec.gateway.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

类型

`object`

必填

`topologyKey`

属性	类型	描述
`labelSelector`	`object`	
`matchLabelKeys`	`array`	
`mismatchLabelKeys`	`array`	
`namespaceSelector`	`object`	
`namespaces`	`array`	
`topologyKey`	`string`	

.spec.gateway.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.gateway.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions

类型

`array`

.spec.gateway.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.gateway.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values

类型

`array`

.spec.gateway.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]

类型

`string`

.spec.gateway.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels

类型

`object`

.spec.gateway.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys

类型

`array`

.spec.gateway.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]

类型

`string`

.spec.gateway.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys

类型

`array`

.spec.gateway.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]

类型

`string`

.spec.gateway.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.gateway.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions

类型

`array`

.spec.gateway.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	

属性	类型	描述
`operator`	`string`	
`values`	`array`	

.spec.gateway.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

类型

`array`

.spec.gateway.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]

类型

`string`

.spec.gateway.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels

类型

`object`

.spec.gateway.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces

类型

`array`

.spec.gateway.placement.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]

类型

`string`

.spec.gateway.placement.podAntiAffinity

类型

`object`

属性	类型	描述
`preferredDuringSchedulingIgnoredDuringExecution`	`array`	
`requiredDuringSchedulingIgnoredDuringExecution`	`array`	

.spec.gateway.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution

类型

`array`

.spec.gateway.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

类型

`object`

必填

`podAffinityTerm` `weight`

属性	类型	描述
`podAffinityTerm`	`object`	
`weight`	`integer`	

.spec.gateway.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm

类型

`object`

必填

`topologyKey`

属性	类型	描述
`labelSelector`	`object`	
`matchLabelKeys`	`array`	
`mismatchLabelKeys`	`array`	
`namespaceSelector`	`object`	
`namespaces`	`array`	
`topologyKey`	`string`	

.spec.gateway.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.gateway.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions

类型

`array`

.spec.gateway.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.gateway.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.lab

elSelector.matchExpressions[].values

类型

`array`

.spec.gateway.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]

类型

`string`

.spec.gateway.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels

类型

`object`

.spec.gateway.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys

类型

`array`

.spec.gateway.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]

类型

`string`

.spec.gateway.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys

类型

`array`

.spec.gateway.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]

类型

`string`

.spec.gateway.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.gateway.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions

类型

`array`

.spec.gateway.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.gateway.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values

类型

`array`

.spec.gateway.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]

类型

`string`

.spec.gateway.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

类型

`object`

.spec.gateway.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

类型

`array`

.spec.gateway.placement.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

类型

`string`

.spec.gateway.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution

类型

`array`

.spec.gateway.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

类型

`object`

必填

`topologyKey`

属性	类型	描述
`labelSelector`	`object`	
`matchLabelKeys`	`array`	
`mismatchLabelKeys`	`array`	
`namespaceSelector`	`object`	
`namespaces`	`array`	
`topologyKey`	`string`	

.spec.gateway.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.gateway.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions

类型

`array`

.spec.gateway.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.gateway.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchE

xpressions[].values

类型

`array`

.spec.gateway.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]

类型

`string`

.spec.gateway.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels

类型

`object`

.spec.gateway.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys

类型

`array`

.spec.gateway.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]

类型

`string`

.spec.gateway.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys

类型

`array`

.spec.gateway.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]

类型

`string`

.spec.gateway.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.gateway.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions

类型

`array`

.spec.gateway.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.gateway.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

类型

`array`

.spec.gateway.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]

类型

`string`

.spec.gateway.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels

类型

`object`

.spec.gateway.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces

类型

`array`

.spec.gateway.placement.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]

类型

`string`

.spec.gateway.placement.tolerations

类型

`array`

.spec.gateway.placement.tolerations[]

类型

`object`

属性	类型	描述
`effect`	`string`	
`key`	`string`	
`operator`	`string`	
`tolerationSeconds`	`integer`	
`value`	`string`	

.spec.gateway.placement.topologySpreadConstraints

类型

`array`

.spec.gateway.placement.topologySpreadConstraints[]

类型

`object`

必填

`maxSkew` `topologyKey` `whenUnsatisfiable`

属性	类型	描述
`labelSelector`	`object`	
`matchLabelKeys`	`array`	
`maxSkew`	`integer`	
`minDomains`	`integer`	
`nodeAffinityPolicy`	`string`	

属性	类型	描述
`nodeTaintsPolicy`	`string`	
`topologyKey`	`string`	
`whenUnsatisfiable`	`string`	

.spec.gateway.placement.topologySpreadConstraints[].labelSelector

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	
`matchLabels`	`object`	

.spec.gateway.placement.topologySpreadConstraints[].labelSelector.matchExpressions

类型

`array`

.spec.gateway.placement.topologySpreadConstraints[].labelSelector.matchExpressions[]

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	
`operator`	`string`	
`values`	`array`	

.spec.gateway.placement.topologySpreadConstraints[].labelSelector.matchExpressions[].values

类型

`array`

.spec.gateway.placement.topologySpreadConstraints[].labelSelector.matchExpressions[].values[]

类型

`string`

.spec.gateway.placement.topologySpreadConstraints[].labelSelector.matchLabels

类型

`object`

.spec.gateway.placement.topologySpreadConstraints[].matchLabelKeys

类型

`array`

.spec.gateway.placement.topologySpreadConstraints[].matchLabelKeys[]

类型

`string`

.spec.gateway.readAffinity

描述

ReadAffinity defines the RGW read affinity policy to optimize the read requests for the RGW clients Note: Only supported from Ceph Tentacle (v20)

类型

`object`

必填

`type`

属性	类型	描述
`type`	`string`	Type defines the RGW ReadAffinity type localize: read from the nearest OSD based on crush location of the RGW client balance: picks a random OSD from the PG's active set default: read from the primary OSD

.spec.gateway.resources

描述

The resource requirements for the rgw pods

类型

`object`

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

.spec.gateway.resources.claims

描述

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

类型

`array`

.spec.gateway.resources.claims[]

描述

ResourceClaim references one entry in PodSpec.ResourceClaims.

类型

`object`

必填

`name`

属性	类型	描述
`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.gateway.resources.limits

描述

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

类型

`object`

.spec.gateway.resources.requests

描述

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/

类型

`object`

.spec.gateway.rgwCommandFlags

描述

RgwCommandFlags sets Ceph RGW config values for the gateway clients that serve this object store. Values are modified at RGW startup, resulting in RGW pod restarts. This feature is intended for advanced users. It allows breaking configurations to be easily applied. Use with caution.

类型

`object`

.spec.gateway.rgwConfig

描述

RgwConfig sets Ceph RGW config values for the gateway clients that serve this object store. Values are modified at runtime without RGW restart. This feature is intended for advanced users. It allows breaking configurations to be easily applied. Use with caution.

类型

`object`

.spec.gateway.rgwConfigFromSecret

描述

RgwConfigFromSecret works exactly like RgwConfig but takes config value from Secret Key reference. Values are modified at runtime without RGW restart. This feature is intended for advanced users. It allows breaking configurations to be easily applied. Use with caution.

类型

`object`

.spec.gateway.service

描述

The configuration related to add/set on each rgw service.

类型

`object`

属性	类型	描述
`annotations`	`object`	The annotations-related configuration to add/set on each rgw service. nullable optional

.spec.gateway.service.annotations

描述

The annotations-related configuration to add/set on each rgw service. nullable optional

类型

`object`

.spec.healthCheck

描述

The RGW health probes

类型

`object`

属性	类型	描述
`readinessProbe`	`object`	ProbeSpec is a wrapper around Probe so it can be enabled or disabled for a Ceph daemon
`startupProbe`	`object`	ProbeSpec is a wrapper around Probe so it can be enabled or disabled for a Ceph daemon

.spec.healthCheck.readinessProbe

描述

ProbeSpec is a wrapper around Probe so it can be enabled or disabled for a Ceph daemon

类型

`object`

属性	类型	描述
`disabled`	`boolean`	Disabled determines whether probe is disable or not
`probe`	`object`	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

.spec.healthCheck.readinessProbe.probe

描述

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

类型

`object`

属性	类型	描述
`exec`	`object`	Exec specifies a command to execute in th
`failureThreshold`	`integer`	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
`grpc`	`object`	GRPC specifies a GRPC HealthCheckReq
`httpGet`	`object`	HTTPGet specifies an HTTP GET request
`initialDelaySeconds`	`integer`	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗
`periodSeconds`	`integer`	How often (in seconds) to perform the prob 10 seconds. Minimum value is 1.
`successThreshold`	`integer`	Minimum consecutive successes for the pr considered successful after having failed. I Must be 1 for liveness and startup. Minimu

属性	类型	描述
`tcpSocket`	`object`	TCPSocket specifies a connection to a TC
`terminationGracePeriodSeconds`	`integer`	
`timeoutSeconds`	`integer`	Number of seconds after which the probe t Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗

.spec.healthCheck.readinessProbe.probe.exec

描述

Exec specifies a command to execute in the container.

类型

`object`

属性	类型	描述
`command`	`array`	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.healthCheck.readinessProbe.probe.exec.command

描述

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

类型

`array`

.spec.healthCheck.readinessProbe.probe.exec.command[]

类型

`string`

.spec.healthCheck.readinessProbe.probe.grpc

描述

GRPC specifies a GRPC HealthCheckRequest.

类型

`object`

必填

`port`

属性	类型	描述
`port`	`integer`	Port number of the gRPC service. Number must be in the range 1 to 65535.
`service`	`string`	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md↗).

属性	类型	描述
		If this is not specified, the default behavior is defined by gRPC.

.spec.healthCheck.readinessProbe.probe.httpGet

描述

HTTPGet specifies an HTTP GET request to perform.

类型

`object`

必填

`port`

属性	类型	描述
`host`	`string`	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
`httpHeaders`	`array`	Custom headers to set in the request. HTTP allows repeated headers.
`path`	`string`	Path to access on the HTTP server.
`port`	`integer|string`	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

属性	类型	描述
`scheme`	`string`	Scheme to use for connecting to the host. Defaults to HTTP.

.spec.healthCheck.readinessProbe.probe.httpGet.httpHeaders

描述

Custom headers to set in the request. HTTP allows repeated headers.

类型

`array`

.spec.healthCheck.readinessProbe.probe.httpGet.httpHeaders[]

描述

HTTPHeader describes a custom header to be used in HTTP probes

类型

`object`

必填

`name` `value`

属性	类型	描述
`name`	`string`	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.

属性	类型	描述
`value`	`string`	The header field value

.spec.healthCheck.readinessProbe.probe.tcpSocket

描述

TCPSocket specifies a connection to a TCP port.

类型

`object`

必填

`port`

属性	类型	描述
`host`	`string`	Optional: Host name to connect to, defaults to the pod IP.
`port`	`integer|string`	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

.spec.healthCheck.startupProbe

描述

ProbeSpec is a wrapper around Probe so it can be enabled or disabled for a Ceph daemon

类型

`object`

属性	类型	描述
`disabled`	`boolean`	Disabled determines whether probe is disable or not
`probe`	`object`	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

.spec.healthCheck.startupProbe.probe

描述

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

类型

`object`

属性	类型	描述
`exec`	`object`	Exec specifies a command to execute in th
`failureThreshold`	`integer`	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
`grpc`	`object`	GRPC specifies a GRPC HealthCheckReq
`httpGet`	`object`	HTTPGet specifies an HTTP GET request

属性	类型	描述
`initialDelaySeconds`	`integer`	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗
`periodSeconds`	`integer`	How often (in seconds) to perform the prob 10 seconds. Minimum value is 1.
`successThreshold`	`integer`	Minimum consecutive successes for the pr considered successful after having failed. I Must be 1 for liveness and startup. Minimu
`tcpSocket`	`object`	TCPSocket specifies a connection to a TC
`terminationGracePeriodSeconds`	`integer`	
`timeoutSeconds`	`integer`	Number of seconds after which the probe t Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗

.spec.healthCheck.startupProbe.probe.exec

描述

Exec specifies a command to execute in the container.

类型

`object`

属性	类型	描述
`command`	`array`	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.healthCheck.startupProbe.probe.exec.command

描述

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

类型

`array`

.spec.healthCheck.startupProbe.probe.exec.command[]

类型

`string`

.spec.healthCheck.startupProbe.probe.grpc

描述

GRPC specifies a GRPC HealthCheckRequest.

类型

`object`

必填

`port`

属性	类型	描述
`port`	`integer`	Port number of the gRPC service. Number must be in the range 1 to 65535.
`service`	`string`	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md↗). If this is not specified, the default behavior is defined by gRPC.

.spec.healthCheck.startupProbe.probe.httpGet

描述

HTTPGet specifies an HTTP GET request to perform.

类型

`object`

必填

`port`

属性	类型	描述
`host`	`string`	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
`httpHeaders`	`array`	Custom headers to set in the request. HTTP allows repeated headers.
`path`	`string`	Path to access on the HTTP server.
`port`	`integer|string`	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
`scheme`	`string`	Scheme to use for connecting to the host. Defaults to HTTP.

.spec.healthCheck.startupProbe.probe.httpGet.httpHeaders

描述

Custom headers to set in the request. HTTP allows repeated headers.

类型

`array`

.spec.healthCheck.startupProbe.probe.httpGet.httpHeaders[]

描述

HTTPHeader describes a custom header to be used in HTTP probes

类型

`object`

必填

`name` `value`

属性	类型	描述
`name`	`string`	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
`value`	`string`	The header field value

.spec.healthCheck.startupProbe.probe.tcpSocket

描述

TCPSocket specifies a connection to a TCP port.

类型

`object`

必填

`port`

属性	类型	描述
`host`	`string`	Optional: Host name to connect to, defaults to the pod IP.

属性	类型	描述
`port`	`integer|string`	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

.spec.hosting

描述

Hosting settings for the object store. A common use case for hosting configuration is to inform Rook of endpoints that support DNS wildcards, which in turn allows virtual host-style bucket addressing.

类型

`object`

属性	类型	描述
`advertiseEndpoint`	`object`	AdvertiseEndpoint is the default endpoint Rook will return for resources dependent on this object store. This endpoint will be returned to CephObjectStoreUsers, Object Bucket Claims, and COSI Buckets/Accesses. By default, Rook returns the endpoint for the object store's Kubernetes service using HTTPS with `gateway.securePort` if it is defined (otherwise, HTTP with `gateway.port`).
`dnsNames`	`array`	A list of DNS host names on which object store gateways will accept client S3 connections. When specified, object store gateways will reject client S3 connections to hostnames that are not present in this list, so include all endpoints. The object store's

属性	类型	描述
		advertiseEndpoint and Kubernetes service endpoint, plus CephObjectZone `customEndpoints` are automatically added to the list but may be set here again if desired. Each DNS name must be valid according RFC-1123. If the DNS name corresponds to an endpoint with DNS wildcard support, do not include the wildcard itself in the list of hostnames. E.g., use "mystore.example.com" instead of "*.mystore.example.com".

.spec.hosting.advertiseEndpoint

描述

AdvertiseEndpoint is the default endpoint Rook will return for resources dependent on this object store. This endpoint will be returned to CephObjectStoreUsers, Object Bucket Claims, and COSI Buckets/Accesses. By default, Rook returns the endpoint for the object store's Kubernetes service using HTTPS with `gateway.securePort` if it is defined (otherwise, HTTP with `gateway.port`).

类型

`object`

必填

`dnsName` `port` `useTls`

属性	类型	描述
`dnsName`	`string`	DnsName is the DNS name (in RFC-1123 format) of the endpoint. If the DNS name corresponds to an endpoint with DNS wildcard support, do not include the wildcard itself in the list of hostnames. E.g., use "mystore.example.com" instead of "*.mystore.example.com".
`port`	`integer`	Port is the port on which S3 connections can be made for this endpoint.
`useTls`	`boolean`	UseTls defines whether the endpoint uses TLS (HTTPS) or not (HTTP).

.spec.hosting.dnsNames

描述

A list of DNS host names on which object store gateways will accept client S3 connections. When specified, object store gateways will reject client S3 connections to hostnames that are not present in this list, so include all endpoints. The object store's advertiseEndpoint and Kubernetes service endpoint, plus CephObjectZone `customEndpoints` are automatically added to the list but may be set here again if desired. Each DNS name must be valid according RFC-1123. If the DNS name corresponds to an endpoint with DNS wildcard support, do not include the wildcard itself in the list of hostnames. E.g., use "mystore.example.com" instead of "*.mystore.example.com".

类型

`array`

.spec.hosting.dnsNames[]

类型

`string`

.spec.metadataPool

描述

The metadata pool settings

类型

`object`

属性	类型	描述
`application`	`string`	The application name to set on the pool. Only expected to be set for rgw pools.
`compressionMode`	`string`	DEPRECATED: use Parameters instead, e.g., Parameters["compression_mode"] = "force" The inline compression mode in Bluestore OSD to set to (options are: none, passive, aggressive, force) Do NOT set a default value for kubebuilder as this will override the Parameters
`crushRoot`	`string`	The root of the crush hierarchy utilized by the pool
`deviceClass`	`string`	The device class the OSD should set to for use in the pool

属性	类型	描述
`enableCrushUpdates`	`boolean`	Allow rook operator to change the pool CRUSH tunables once the pool is created
`enableRBDStats`	`boolean`	EnableRBDStats is used to enable gathering of statistics for all RBD images in the pool
`erasureCoded`	`object`	The erasure code settings
`failureDomain`	`string`	The failure domain: osd/host/(region or zone if available) - technically also any type in the crush map
`mirroring`	`object`	The mirroring settings
`parameters`	`object`	Parameters is a list of properties to enable on a given pool
`quotas`	`object`	The quota settings
`replicated`	`object`	The replication settings
`statusCheck`	`object`	The mirroring statusCheck

.spec.metadataPool.erasureCoded

描述

The erasure code settings

类型

`object`

必填

`codingChunks` `dataChunks`

属性	类型	描述
`algorithm`	`string`	The algorithm for erasure coding. If absent, defaults to the plugin specified in osd_pool_default_erasure_code_profile.
`codingChunks`	`integer`	Number of coding chunks per object in an erasure coded storage pool (required for erasure-coded pool type). This is the number of OSDs that can be lost simultaneously before data cannot be recovered.
`dataChunks`	`integer`	Number of data chunks per object in an erasure coded storage pool (required for erasure-coded pool type). The number of chunks required to recover an object when any single OSD is lost is the same as dataChunks so be aware that the larger the number of data chunks, the higher the cost of recovery.

.spec.metadataPool.mirroring

描述

The mirroring settings

类型

`object`

属性	类型	描述
`enabled`	`boolean`	Enabled whether this pool is mirrored or not
`mode`	`string`	Mode is the mirroring mode: pool, image or init-only.
`peers`	`object`	Peers represents the peers spec
`snapshotSchedules`	`array`	SnapshotSchedules is the scheduling of snapshot for mirrored images/pools

.spec.metadataPool.mirroring.peers

描述

Peers represents the peers spec

类型

`object`

属性	类型	描述
`secretNames`	`array`	SecretNames represents the Kubernetes Secret names to add rbd-mirror or cephfs-mirror peers

.spec.metadataPool.mirroring.peers.secretNames

描述

SecretNames represents the Kubernetes Secret names to add rbd-mirror or cephfs-mirror peers

类型

`array`

.spec.metadataPool.mirroring.peers.secretNames[]

类型

`string`

.spec.metadataPool.mirroring.snapshotSchedules

描述

SnapshotSchedules is the scheduling of snapshot for mirrored images/pools

类型

`array`

.spec.metadataPool.mirroring.snapshotSchedules[]

描述

SnapshotScheduleSpec represents the snapshot scheduling settings of a mirrored pool

类型

`object`

属性	类型	描述
`interval`	`string`	Interval represent the periodicity of the snapshot.

属性	类型	描述
`path`	`string`	Path is the path to snapshot, only valid for CephFS
`startTime`	`string`	StartTime indicates when to start the snapshot

.spec.metadataPool.parameters

描述

Parameters is a list of properties to enable on a given pool

类型

`object`

.spec.metadataPool.quotas

描述

The quota settings

类型

`object`

属性	类型	描述
`maxBytes`	`integer`	MaxBytes represents the quota in bytes Deprecated in favor of MaxSize
`maxObjects`	`integer`	MaxObjects represents the quota in objects

属性	类型	描述
`maxSize`	`string`	MaxSize represents the quota in bytes as a string

.spec.metadataPool.replicated

描述

The replication settings

类型

`object`

必填

`size`

属性	类型	描述
`hybridStorage`	`object`	HybridStorage represents hybrid storage tier settings
`replicasPerFailureDomain`	`integer`	ReplicasPerFailureDomain the number of replica in the specified failure domain
`requireSafeReplicaSize`	`boolean`	RequireSafeReplicaSize if false allows you to set replica 1
`size`	`integer`	Size - Number of copies per object in a replicated storage pool, including the object itself (required for replicated pool type)

属性	类型	描述
`subFailureDomain`	`string`	SubFailureDomain the name of the sub-failure domain
`targetSizeRatio`	`number`	TargetSizeRatio gives a hint (%) to Ceph in terms of expected consumption of the total cluster capacity

.spec.metadataPool.replicated.hybridStorage

描述

HybridStorage represents hybrid storage tier settings

类型

`object`

必填

`primaryDeviceClass` `secondaryDeviceClass`

属性	类型	描述
`primaryDeviceClass`	`string`	PrimaryDeviceClass represents high performance tier (for example SSD or NVME) for Primary OSD
`secondaryDeviceClass`	`string`	SecondaryDeviceClass represents low performance tier (for example HDDs) for remaining OSDs

.spec.metadataPool.statusCheck

描述

The mirroring statusCheck

类型

`object`

属性	类型	描述
`mirror`	`object`	HealthCheckSpec represents the health check of an object store bucket

.spec.metadataPool.statusCheck.mirror

描述

HealthCheckSpec represents the health check of an object store bucket

类型

`object`

属性	类型	描述
`disabled`	`boolean`	
`interval`	`string`	Interval is the internal in second or minute for the health check to run like 60s for 60 seconds
`timeout`	`string`	

.spec.protocols

描述

The protocol specification

类型

`object`

属性	类型	描述
`enableAPIs`	`array`	Represents RGW 'rgw_enable_apis' config option. See: https://docs.ceph.com/en/reef/radosgw/config-ref/#confval-rgw_enable_apis↗ If no value provided then all APIs will be enabled: s3, s3website, swift, swift_auth, admin, sts, iam, notifications If enabled APIs are set, all remaining APIs will be disabled. This option overrides S3.Enabled value.
`s3`	`object`	The spec for S3
`swift`	`object`	The spec for Swift

.spec.protocols.enableAPIs

描述

Represents RGW 'rgw_enable_apis' config option. See: https://docs.ceph.com/en/reef/radosgw/config-ref/#confval-rgw_enable_apis If no value provided then all APIs will be enabled: s3, s3website, swift, swift_auth, admin, sts, iam, notifications If enabled APIs are set, all remaining APIs will be disabled. This option overrides S3.Enabled value.

类型

`array`

.spec.protocols.enableAPIs[]

类型

`string`

.spec.protocols.s3

描述

The spec for S3

类型

`object`

属性	类型	描述
`authUseKeystone`	`boolean`	Whether to use Keystone for authentication. This option maps directly to the rgw_s3_auth_use_keystone option. Enabling it allows generating S3 credentials via an OpenStack API call, see the docs. If not given, the defaults of the corresponding RGW option apply.
`enabled`	`boolean`	Deprecated: use protocol.enableAPIs instead. Whether to enable S3. This defaults to true (even if protocols.s3 is not present in the CRD). This maintains backwards compatibility – by default S3 is enabled.

.spec.protocols.swift

描述

The spec for Swift

类型

`object`

属性	类型	描述
`accountInUrl`	`boolean`	Whether or not the Swift account name should be included in the Swift API URL. If set to false (the default), then the Swift API will listen on a URL formed like http://host:port/<rgw_swift_url_prefix>/v1. If set to true, the Swift API URL will be http://host:port/<rgw_swift_url_prefix>/v1/AUTH_<account_name>. You must set this option to true (and update the Keystone service catalog) if you want radosgw to support publicly-readable containers and temporary URLs.
`urlPrefix`	`string`	The URL prefix for the Swift API, to distinguish it from the S3 API endpoint. The default is swift, which makes the Swift API available at the URL http://host:port/swift/v1 (or http://host:port/swift/v1/AUTH_%(tenant_id)s if rgw swift account in url is enabled).
`versioningEnabled`	`boolean`	Enables the Object Versioning of OpenStack Object Storage API. This allows clients to put the X-Versions-Location attribute on containers that should be versioned.

.spec.security

描述

Security represents security settings

类型

`object`

属性	类型	描述
`keyRotation`	`object`	KeyRotation defines options for Key Rotation.
`kms`	`object`	KeyManagementService is the main Key Management option
`s3`	`object`	The settings for supporting AWS-SSE:S3 with RGW

.spec.security.keyRotation

描述

KeyRotation defines options for Key Rotation.

类型

`object`

属性	类型	描述
`enabled`	`boolean`	Enabled represents whether the key rotation is enabled.
`schedule`	`string`	Schedule represents the cron schedule for key rotation.

.spec.security.kms

描述

KeyManagementService is the main Key Management option

类型

`object`

属性	类型	描述
`connectionDetails`	`object`	ConnectionDetails contains the KMS connection details (address, port etc)
`tokenSecretName`	`string`	TokenSecretName is the kubernetes secret containing the KMS token

.spec.security.kms.connectionDetails

描述

ConnectionDetails contains the KMS connection details (address, port etc)

类型

`object`

.spec.security.s3

描述

The settings for supporting AWS-SSE:S3 with RGW

类型

`object`

属性	类型	描述
`connectionDetails`	`object`	ConnectionDetails contains the KMS connection details (address, port etc)

属性	类型	描述
`tokenSecretName`	`string`	TokenSecretName is the kubernetes secret containing the KMS token

.spec.security.s3.connectionDetails

描述

ConnectionDetails contains the KMS connection details (address, port etc)

类型

`object`

.spec.sharedPools

描述

The pool information when configuring RADOS namespaces in existing pools.

类型

`object`

属性	类型	描述
`dataPoolName`	`string`	The data pool used for creating RAD object store
`metadataPoolName`	`string`	The metadata pool used for creating the object store
`poolPlacements`	`array`	PoolPlacements control which Pools particular RGW bucket. Once PoolPl

属性	类型	描述
		RGW client will be able to associate bucket by providing "" during s3 buck Policy" header during swift container https://docs.ceph.com/en/latest/rados targets ↗ PoolPlacement with name: default pool if no option is provided d default placement is not provided, spec.sharedPools.dataPoolName an spec.sharedPools.MetadataPoolNam pools. If spec.sharedPools are also e (spec.dataPool and spec.metadataP defaults.
`preserveRadosNamespaceDataOnDelete`	`boolean`	Whether the RADOS namespaces sl deletion of the object store

.spec.sharedPools.poolPlacements

描述

PoolPlacements control which Pools are associated with a particular RGW bucket. Once PoolPlacements are defined, RGW client will be able to associate pool with ObjectStore bucket by providing "<LocationConstraint>" during s3 bucket creation or "X-Storage-Policy" header during swift container creation. See: https://docs.ceph.com/en/latest/radosgw/placement/#placement-targets PoolPlacement with name: "default" will be used as a default pool if no option is provided during bucket creation. If default placement is not provided, spec.sharedPools.dataPoolName and spec.sharedPools.MetadataPoolName will be used as default pools. If spec.sharedPools are also empty, then RGW pools (spec.dataPool and spec.metadataPool) will be used as defaults.

类型

`array`

.spec.sharedPools.poolPlacements[]

类型

`object`

必填

`dataPoolName` `metadataPoolName` `name`

属性	类型	描述
`dataNonECPoolName`	`string`	The data pool used to store ObjectStore data that cannot use erasure coding (ex: multi-part uploads). If dataPoolName is not erasure coded, then there is no need for dataNonECPoolName.
`dataPoolName`	`string`	The data pool used to store ObjectStore objects data.
`default`	`boolean`	Sets given placement as default. Only one placement in the list can be marked as default. Default is false.
`metadataPoolName`	`string`	The metadata pool used to store ObjectStore bucket index.
`name`	`string`	Pool placement name. Name can be arbitrary. Placement with name "default" will be used as default.

属性	类型	描述
`storageClasses`	`array`	StorageClasses can be selected by user to override dataPoolName during object creation. Each placement has default STANDARD StorageClass pointing to dataPoolName. This list allows defining additional StorageClasses on top of default STANDARD storage class.

.spec.sharedPools.poolPlacements[].storageClasses

描述

StorageClasses can be selected by user to override dataPoolName during object creation. Each placement has default STANDARD StorageClass pointing to dataPoolName. This list allows defining additional StorageClasses on top of default STANDARD storage class.

类型

`array`

.spec.sharedPools.poolPlacements[].storageClasses[]

类型

`object`

必填

`dataPoolName` `name`

属性	类型	描述
`dataPoolName`	`string`	DataPoolName is the data pool used to store ObjectStore objects data.

属性	类型	描述
`name`	`string`	Name is the StorageClass name. Ceph allows arbitrary name for StorageClasses, however most clients/libs insist on AWS names so it is recommended to use one of the valid x-amz-storage-class values for better compatibility: REDUCED_REDUNDANCY | STANDARD_IA | ONEZONE_IA | INTELLIGENT_TIERING | GLACIER | DEEP_ARCHIVE | OUTPOSTS | GLACIER_IR | SNOW | EXPRESS_ONEZONE See AWS docs: https://aws.amazon.com/de/s3/storage-classes/ ↗

.spec.zone

描述

The multisite info

类型

`object`

必填

`name`

属性	类型	描述
`name`	`string`	CephObjectStoreZone name this CephObjectStore is part of

.status

描述

ObjectStoreStatus represents the status of a Ceph Object Store resource

类型

`object`

属性	类型	描述
`cephx`	`object`	
`conditions`	`array`	
`endpoints`	`object`	
`info`	`object`	
`message`	`string`	
`observedGeneration`	`integer`	ObservedGeneration is the latest generation observed by the controller.
`phase`	`string`	ConditionType represent a resource's status
`replicas`	`integer`	
`selector`	`string`	

.status.cephx

类型

`object`

属性	类型	描述
`daemon`	`object`	Daemon shows the CephX key status for local Ceph daemons associated with this resources.

.status.cephx.daemon

描述

Daemon shows the CephX key status for local Ceph daemons associated with this resources.

类型

`object`

属性	类型	描述
`keyCephVersion`	`string`	KeyCephVersion reports the Ceph version that created the current generation's keys. This is same string format as reported by `CephCluster.status.version.version` to allow them to be compared. E.g., `20.2.0-0`. For all newly-created resources, this field set to the version of Ceph that created the key. The special value "Uninitialized" indicates that keys are being created for the first time. An empty string indicates that the version is unknown, as expected in brownfield deployments.
`keyGeneration`	`integer`	KeyGeneration represents the CephX key generation for the last successful reconcile. For all newly-created resources, this field is set to `1`. When keys are rotated due to any rotation policy, the generation is incremented or updated to the configured policy generation. Generation `0` indicates that keys existed prior to the implementation of key tracking.

.status.conditions

类型

`array`

.status.conditions[]

描述

Condition represents a status condition on any Rook-Ceph Custom Resource.

类型

`object`

属性	类型	描述
`lastHeartbeatTime`	`string`	
`lastTransitionTime`	`string`	
`message`	`string`	
`reason`	`string`	ConditionReason is a reason for a condition
`status`	`string`	
`type`	`string`	ConditionType represent a resource's status

.status.endpoints

类型

`object`

属性	类型	描述
`insecure`	`array`	
`secure`	`array`	

.status.endpoints.insecure

类型

`array`

.status.endpoints.insecure[]

类型

`string`

.status.endpoints.secure

类型

`array`

.status.endpoints.secure[]

类型

`string`

.status.info

类型

`object`

API 端点

可用的 API 端点如下：

- `/apis/ceph.rook.io/v1/namespaces/{namespace}/cephobjectstores`
  - `DELETE`: delete collection of CephObjectStore
  - `GET`: list objects of kind CephObjectStore

- `POST` : create a new CephObjectStore

- `/apis/ceph.rook.io/v1/namespaces/{namespace}/cephobjectstores/{name}`
  - `DELETE` : delete the specified CephObjectStore
  - `GET` : read the specified CephObjectStore
  - `PATCH` : partially update the specified CephObjectStore
  - `PUT` : replace the specified CephObjectStore
- `/apis/ceph.rook.io/v1/namespaces/{namespace}/cephobjectstores/{name}/status`
  - `GET` : read status of the specified CephObjectStore
  - `PATCH` : partially update status of the specified CephObjectStore
  - `PUT` : replace status of the specified CephObjectStore
- `/apis/ceph.rook.io/v1/namespaces/{namespace}/cephobjectstores/{name}/scale`
  - `GET` : read scale of the specified CephObjectStore
  - `PATCH` : partially update scale of the specified CephObjectStore
  - `PUT` : replace scale of the specified CephObjectStore

/apis/ceph.rook.io/v1/namespaces/{namespace}/cephobjectstores

HTTP 方法

`DELETE`

描述

delete collection of CephObjectStore

HTTP 响应

HTTP 状态码	响应体
200 - OK	`Status` schema
401 - Unauthorized	Empty

HTTP 方法

`GET`

描述

list objects of kind CephObjectStore

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephObjectStoreList` schema
401 - Unauthorized	Empty

HTTP 方法

`POST`

描述

create a new CephObjectStore

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	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.

参数	类型	描述
		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.

请求体参数

参数	类型	描述
`body`	`CephObjectStore` schema	`application/json` formatted

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephObjectStore` schema
201 - Created	`CephObjectStore` schema
202 - Accepted	`CephObjectStore` schema
401 - Unauthorized	Empty

/apis/ceph.rook.io/v1/namespaces/{namespace}/cephobjectstores/{name}

HTTP 方法

`DELETE`

描述

delete the specified CephObjectStore

查询参数

参数	类型	描述
`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 响应

HTTP 状态码	响应体
200 - OK	`Status` schema
202 - Accepted	`Status` schema
401 - Unauthorized	Empty

HTTP 方法

`GET`

描述

read the specified CephObjectStore

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephObjectStore` schema
401 - Unauthorized	Empty

HTTP 方法

`PATCH`

描述

partially update the specified CephObjectStore

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	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.

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephObjectStore` schema
401 - Unauthorized	Empty

HTTP 方法

`PUT`

描述

replace the specified CephObjectStore

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	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.

请求体参数

参数	类型	描述
`body`	`CephObjectStore` schema	`application/json` formatted

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephObjectStore` schema

HTTP 状态码	响应体
201 - Created	`CephObjectStore` schema
401 - Unauthorized	Empty

/apis/ceph.rook.io/v1/namespaces/{namespace}/cephobjectstores/{name}/status

HTTP 方法

`GET`

描述

read status of the specified CephObjectStore

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephObjectStore` schema
401 - Unauthorized	Empty

HTTP 方法

`PATCH`

描述

partially update status of the specified CephObjectStore

查询参数

参数	类型	描述
`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

参数	类型	描述
`fieldValidation`	`string`	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.

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephObjectStore` schema
401 - Unauthorized	Empty

HTTP 方法

`PUT`

描述

replace status of the specified CephObjectStore

查询参数

参数	类型	描述
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun

参数	类型	描述
		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
`fieldValidation`	`string`	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.

请求体参数

参数	类型	描述
`body`	`CephObjectStore` schema	`application/json` formatted

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephObjectStore` schema
201 - Created	`CephObjectStore` schema
401 - Unauthorized	Empty

/apis/ceph.rook.io/v1/namespaces/{namespace}/cephobjectstores/{name}/scale

HTTP 方法

`GET`

描述

read scale of the specified CephObjectStore

HTTP 响应

HTTP 状态码	响应体
200 - OK	`Scale` schema
401 - Unauthorized	Empty

HTTP 方法

`PATCH`

描述

partially update scale of the specified CephObjectStore

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	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

参数	类型	描述
		warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP 方法

`PUT`

描述

replace scale of the specified CephObjectStore

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	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.

参数	类型	描述
		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.

请求体参数

参数	类型	描述
`body`	`Scale` schema	`application/json` formatted

CephObjectStoreUser [cephobjectstoreusers.ceph.rook.io/v1]

描述

CephObjectStoreUser represents a Ceph Object Store Gateway User

类型

`object`

必填

`metadata` `spec`

规格

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

属性	类型	描述
`kind`	`string`	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`metadata`	`ObjectMeta`	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
`spec`	`object`	ObjectStoreUserSpec represent the spec of an Objectstoreuser
`status`	`object`	ObjectStoreUserStatus represents the status Ceph Object Store Gateway User

.spec

描述

ObjectStoreUserSpec represent the spec of an Objectstoreuser

类型

`object`

属性	类型	描述
`capabilities`	`object`	Additional admin-level capabilities for the Ceph object store user
`clusterNamespace`	`string`	The namespace where the parent CephCluster and CephObjectStore are found
`displayName`	`string`	The display name for the ceph user.
`keys`	`array`	Allows specifying credentials for the user. If not provided, the operator will generate them.
`quotas`	`object`	ObjectUserQuotaSpec can be used to set quotas for the object store user to limit their usage. See the Ceph docs↗ for more
`store`	`string`	The store the user will be created in

.spec.capabilities

描述

Additional admin-level capabilities for the Ceph object store user

类型

`object`

属性	类型	描述
`amz-cache`	`string`	Add capabilities for user to send request to RGW Cache API header. Documented in https://docs.ceph.com/en/latest/radosgw/rgw-cache/#cache-api ↗
`bilog`	`string`	Add capabilities for user to change bucket index logging. Documented in https://docs.ceph.com/en/latest/radosgw/admin/?#add-remove-admin-capabilities ↗
`bucket`	`string`	Admin capabilities to read/write Ceph object store buckets. Documented in https://docs.ceph.com/en/latest/radosgw/admin/?#add-remove-admin-capabilities ↗
`buckets`	`string`	Admin capabilities to read/write Ceph object store buckets. Documented in https://docs.ceph.com/en/latest/radosgw/admin/?#add-remove-admin-capabilities ↗
`datalog`	`string`	Add capabilities for user to change data logging. Documented in https://docs.ceph.com/en/latest/radosgw/admin/?#add-remove-admin-capabilities ↗
`info`	`string`	Admin capabilities to read/write information about the user. Documented in

属性	类型	描述
		https://docs.ceph.com/en/latest/radosgw/admin/?#add-remove-admin-capabilities ↗
`mdlog`	`string`	Add capabilities for user to change metadata logging. Documented in https://docs.ceph.com/en/latest/radosgw/admin/?#add-remove-admin-capabilities ↗
`metadata`	`ObjectMeta`	Admin capabilities to read/write Ceph object store metadata. Documented in https://docs.ceph.com/en/latest/radosgw/admin/?#add-remove-admin-capabilities ↗
`oidc-provider`	`string`	Add capabilities for user to change oidc provider. Documented in https://docs.ceph.com/en/latest/radosgw/admin/?#add-remove-admin-capabilities ↗
`ratelimit`	`string`	Add capabilities for user to set rate limiter for user and bucket. Documented in https://docs.ceph.com/en/latest/radosgw/admin/?#add-remove-admin-capabilities ↗
`roles`	`string`	Admin capabilities to read/write roles for user. Documented in https://docs.ceph.com/en/latest/radosgw/admin/?#add-remove-admin-capabilities ↗

属性	类型	描述
`usage`	`string`	Admin capabilities to read/write Ceph object store usage. Documented in https://docs.ceph.com/en/latest/radosgw/admin/?#add-remove-admin-capabilities ↗
`user`	`string`	Admin capabilities to read/write Ceph object store users. Documented in https://docs.ceph.com/en/latest/radosgw/admin/?#add-remove-admin-capabilities ↗
`user-policy`	`string`	Add capabilities for user to change user policies. Documented in https://docs.ceph.com/en/latest/radosgw/admin/?#add-remove-admin-capabilities ↗
`users`	`string`	Admin capabilities to read/write Ceph object store users. Documented in https://docs.ceph.com/en/latest/radosgw/admin/?#add-remove-admin-capabilities ↗
`zone`	`string`	Admin capabilities to read/write Ceph object store zones. Documented in https://docs.ceph.com/en/latest/radosgw/admin/?#add-remove-admin-capabilities ↗

.spec.keys

描述

Allows specifying credentials for the user. If not provided, the operator will generate them.

类型

`array`

.spec.keys[]

描述

ObjectUserKey defines a set of rgw user access credentials to be retrieved from secret resources.

类型

`object`

属性	类型	描述
`accessKeyRef`	`object`	Secret key selector for the access_key (commonly referred to as AWS_ACCESS_KEY_ID).
`secretKeyRef`	`object`	Secret key selector for the secret_key (commonly referred to as AWS_SECRET_ACCESS_KEY).

.spec.keys[].accessKeyRef

描述

Secret key selector for the access_key (commonly referred to as AWS_ACCESS_KEY_ID).

类型

`object`

必填

`key`

属性	类型	描述
`key`	`string`	The key of the secret to select from. Must be a valid secret key.
`name`	`string`	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`optional`	`boolean`	Specify whether the Secret or its key must be defined

.spec.keys[].secretKeyRef

描述

Secret key selector for the secret_key (commonly referred to as AWS_SECRET_ACCESS_KEY).

类型

`object`

必填

`key`

属性	类型	描述
`key`	`string`	The key of the secret to select from. Must be a valid secret key.

属性	类型	描述
`name`	`string`	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`optional`	`boolean`	Specify whether the Secret or its key must be defined

.spec.quotas

描述

ObjectUserQuotaSpec can be used to set quotas for the object store user to limit their usage. See the [Ceph docs](https://docs.ceph.com/en/latest/radosgw/admin/?#quota-management) for more

类型

`object`

属性	类型	描述
`maxBuckets`	`integer`	Maximum bucket limit for the ceph user
`maxObjects`	`integer`	Maximum number of objects across all the user's buckets
`maxSize`	`integer|string`	Maximum size limit of all objects across all the user's bucket See

属性	类型	描述
		https://pkg.go.dev/k8s.io/apimachinery/pkg/api/resource#Qu ↗ for more info.

.status

描述

ObjectStoreUserStatus represents the status Ceph Object Store Gateway User

类型

`object`

属性	类型	描述
`info`	`object`	
`keys`	`array`	
`observedGeneration`	`integer`	ObservedGeneration is the latest generation observed by the controller.
`phase`	`string`	

.status.info

类型

`object`

.status.keys

类型

`array`

.status.keys[]

类型

`object`

属性	类型	描述
`name`	`string`	name is unique within a namespace to reference a secret resource.
`namespace`	`string`	namespace defines the space within which the secret name must be unique.
`resourceVersion`	`string`	
`uid`	`string`	UID is a type that holds unique ID values, including UUIDs. Because we don't ONLY use UUIDs, this is an alias to string. Being a type captures intent and helps make sure that UIDs and names do not get conflated.

API 端点

可用的 API 端点如下：

- `/apis/ceph.rook.io/v1/namespaces/{namespace}/cephobjectstoreusers`
  - `DELETE`: delete collection of CephObjectStoreUser
  - `GET`: list objects of kind CephObjectStoreUser
  - `POST`: create a new CephObjectStoreUser
- `/apis/ceph.rook.io/v1/namespaces/{namespace}/cephobjectstoreusers/{name}`

- `DELETE` : delete the specified CephObjectStoreUser
- `GET` : read the specified CephObjectStoreUser
- `PATCH` : partially update the specified CephObjectStoreUser
- `PUT` : replace the specified CephObjectStoreUser

- `/apis/ceph.rook.io/v1/namespaces/{namespace}/cephobjectstoreusers/{name}/status`
  - `GET` : read status of the specified CephObjectStoreUser
  - `PATCH` : partially update status of the specified CephObjectStoreUser
  - `PUT` : replace status of the specified CephObjectStoreUser

/apis/ceph.rook.io/v1/namespaces/{namespace}/cephobjectstoreusers

HTTP 方法

`DELETE`

描述

delete collection of CephObjectStoreUser

HTTP 响应

HTTP 状态码	响应体
200 - OK	`Status` schema
401 - Unauthorized	Empty

HTTP 方法

`GET`

描述

list objects of kind CephObjectStoreUser

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephObjectStoreUserList` schema
401 - Unauthorized	Empty

HTTP 方法

`POST`

描述

create a new CephObjectStoreUser

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	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.

请求体参数

参数	类型	描述
`body`	`CephObjectStoreUser` schema	`application/json` formatted

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephObjectStoreUser` schema
201 - Created	`CephObjectStoreUser` schema
202 - Accepted	`CephObjectStoreUser` schema
401 - Unauthorized	Empty

/apis/ceph.rook.io/v1/namespaces/{namespace}/cephobjectstoreusers/{name}

HTTP 方法

`DELETE`

描述

delete the specified CephObjectStoreUser

查询参数

参数	类型	描述
`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 响应

HTTP 状态码	响应体
200 - OK	`Status` schema
202 - Accepted	`Status` schema
401 - Unauthorized	Empty

HTTP 方法

`GET`

描述

read the specified CephObjectStoreUser

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephObjectStoreUser` schema
401 - Unauthorized	Empty

HTTP 方法

`PATCH`

描述

partially update the specified CephObjectStoreUser

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore:

参数	类型	描述
		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.

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephObjectStoreUser` schema
401 - Unauthorized	Empty

HTTP 方法

`PUT`

描述

replace the specified CephObjectStoreUser

查询参数

参数	类型	描述
`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

参数	类型	描述
`fieldValidation`	`string`	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.

请求体参数

参数	类型	描述
`body`	`CephObjectStoreUser` schema	`application/json` formatted

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephObjectStoreUser` schema
201 - Created	`CephObjectStoreUser` schema
401 - Unauthorized	Empty

/apis/ceph.rook.io/v1/namespaces/{namespace}/cephobjectstoreusers/{name}/status

HTTP 方法

`GET`

描述

read status of the specified CephObjectStoreUser

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephObjectStoreUser` schema
401 - Unauthorized	Empty

HTTP 方法

`PATCH`

描述

partially update status of the specified CephObjectStoreUser

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	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.

参数	类型	描述
		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 响应

HTTP 状态码	响应体
200 - OK	`CephObjectStoreUser` schema
401 - Unauthorized	Empty

HTTP 方法

`PUT`

描述

replace status of the specified CephObjectStoreUser

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	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,

参数	类型	描述
		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.

请求体参数

参数	类型	描述
`body`	`CephObjectStoreUser` schema	`application/json` formatted

HTTP 响应

HTTP 状态码	响应体
200 - OK	`CephObjectStoreUser` schema
201 - Created	`CephObjectStoreUser` schema
401 - Unauthorized	Empty