GitOps APIs

Core	Application	Application

Core

/gitops/v1/gql

`get` GraphQL endpoint for gitops resources

参数

- `Authorization` (*in header*): `string`
  Given Bearer token will use this as authorization for the API

响应

- `200` : success

`post` GraphQL endpoint for gitops resources

参数

- `Authorization` (*in header*): `string`
  Given Bearer token will use this as authorization for the API

响应

- `200` : success

/gitops/v1/project/{project}/unifiedapps

`get` List application and applicationsets in a project

参数

- `Authorization` (*in header*): `string`
  Given Bearer token will use this as authorization for the API
- `project` (*in path*): `string` required
  project name
- `filter` (*in query*): `string`
  list filter
- `type` (*in query*): `string`
  app type

响应

- `200` UnstructuredList: success

/gitops/v1/project/{project}/repository/apps

`get` Get directories of a repository

参数

- `Authorization` (*in header*): `string`
  Given Bearer token will use this as authorization for the API
- `repoURL` (*in query*): `string`
  repository url
- `revision` (*in query*): `string`
  Revision reference
- `appName` (*in query*): `string`
  name of application

- `appProject` (*in query*): `string`
  name of AppProject

响应

- `200` RepoDirectoryList: success

/gitops/v1/project/{project}/repository/refs

`get` Get references in a repository

参数

- `Authorization` (*in header*): `string`
  Given Bearer token will use this as authorization for the API
- `repoURL` (*in query*): `string`
  repository url

响应

- `200` RepoRefs: success

UnstructuredList

- `Object` : `object`
- `items` : `[]Unstructured`

Unstructured

- `Object` : `object`

RepoDirectoryList

- `items` : `[]RepoDirectory`

RepoDirectory

- `path` : `string`
- `type` : `string`

RepoRefs

- `branches` : `[]string`
- `tags` : `[]string`

Application

/gitops/v1/project/{project}/applications

`get` List applications in a project

参数

- `Authorization` (*in header*): `string`
  Given Bearer token will use this as authorization for the API
- `project` (*in path*): `string` required
  project name
- `name` (*in path*): `string` required
  app name

响应

- `200` ApplicationList: success

`post` Create applications in a project

参数

- `Authorization` (*in header*): `string`
  Given Bearer token will use this as authorization for the API
- `project` (*in path*): `string` required

project name

- `name` (*in path*): `string` required
  app name

响应

- `201` Application: success

/gitops/v1/project/{project}/applications/{name}

`get` Get an application in a project

参数

- `Authorization` (*in header*): `string`
  Given Bearer token will use this as authorization for the API
- `project` (*in path*): `string` required
  project name
- `name` (*in path*): `string` required
  app name

响应

- `200` Application: success

`put` Update an application in a project

参数

- `Authorization` (*in header*): `string`
  Given Bearer token will use this as authorization for the API
- `project` (*in path*): `string` required
  project name
- `name` (*in path*): `string` required

app name

响应

- `200` Application: success

`delete` Delete an application in an project

参数

- `Authorization` (*in header*): `string`
  Given Bearer token will use this as authorization for the API
- `project` (*in path*): `string` required
  project name
- `name` (*in path*): `string` required
  app name

响应

- `204` : success

/gitops/v1/project/{project}/applications/{name}/events

`get` Query events of an applications

参数

- `Authorization` (*in header*): `string`
  Given Bearer token will use this as authorization for the API
- `project` (*in path*): `string` required
  project name
- `name` (*in path*): `string` required

app name

- `start_time` (*in query*): `string`
  start time
- `end_time` (*in query*): `string`
  end time
- `reason` (*in query*): `string`
  reason
- `kind` (*in query*): `string`
  kind
- `name` (*in query*): `string`
  name
- `event_type` (*in query*): `string`
  event type
- `pageno` (*in query*): `integer`
  page number
- `size` (*in query*): `size`
  size

响应

- `200` : success

/gitops/v1/project/{project}/applications/{name}/refresh

`post` Refresh application in an project

参数

- `Authorization` (*in header*): `string`
  Given Bearer token will use this as authorization for the API
- `project` (*in path*): `string` required

project name

- `name` (*in path*): `string` required
  app name

响应

- `204` : success

/gitops/v1/project/{project}/applications/{name}/sync

`post` Sync resources of application in an project

参数

- `Authorization` (*in header*): `string`
  Given Bearer token will use this as authorization for the API
- `project` (*in path*): `string` required
  project name
- `name` (*in path*): `string` required
  app name

响应

- `204` : success

ApplicationList

- `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 ↗

- `items` : `[]Application`
- `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` : ListMeta
  ListMeta describes metadata that synthetic resources must have, including lists and various status objects. A resource may have only one of {ObjectMeta, ListMeta}.

Application

- `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.
- `operation` : Operation
- `spec` : ApplicationSpec
- `status` : ApplicationStatus

Operation

- `info` : `[]Info`

- `initiatedBy` : OperationInitiator
- `retry` : RetryStrategy
- `sync` : SyncOperation

Info

- `name` : `string`
- `value` : `string`

OperationInitiator

- `automated` : `boolean`
- `username` : `string`

RetryStrategy

- `backoff` : Backoff
- `limit` : `integer`

Backoff

- `duration` : `string`
- `factor` : `integer`
- `maxDuration` : `string`

SyncOperation

- `autoHealAttemptsCount` : `integer`

- `dryRun` : `boolean`
- `manifests` : `[]string`
- `prune` : `boolean`
- `resources` : `[]SyncOperationResource`
- `revision` : `string`
- `revisions` : `[]string`
- `source` : ApplicationSource
- `sources` : `[]ApplicationSource`
- `syncOptions` : `[]string`
- `syncStrategy` : SyncStrategy

SyncOperationResource

- `group` : `string`
- `kind` : `string`
- `name` : `string`
- `namespace` : `string`

ApplicationSource

- `chart` : `string`
- `directory` : ApplicationSourceDirectory
- `helm` : ApplicationSourceHelm
- `kustomize` : ApplicationSourceKustomize
- `name` : `string`
- `path` : `string`
- `plugin` : ApplicationSourcePlugin
- `ref` : `string`

- `repoURL` : `string`
- `targetRevision` : `string`

ApplicationSourceDirectory

- `exclude` : `string`
- `include` : `string`
- `jsonnet` : ApplicationSourceJsonnet
- `recurse` : `boolean`

ApplicationSourceJsonnet

- `extVars` : `[]JsonnetVar`
- `libs` : `[]string`
- `tlas` : `[]JsonnetVar`

JsonnetVar

- `code` : `boolean`
- `name` : `string`
- `value` : `string`

ApplicationSourceHelm

- `apiVersions` : `[]string`
- `fileParameters` : `[]HelmFileParameter`
- `ignoreMissingValueFiles` : `boolean`
- `kubeVersion` : `string`

- `namespace` : `string`
- `parameters` : `[]HelmParameter`
- `passCredentials` : `boolean`
- `releaseName` : `string`
- `skipCrds` : `boolean`
- `skipSchemaValidation` : `boolean`
- `skipTests` : `boolean`
- `valueFiles` : `[]string`
- `values` : `string`
- `valuesObject` : `string`
- `version` : `string`

HelmFileParameter

- `name` : `string`
- `path` : `string`

HelmParameter

- `forceString` : `boolean`
- `name` : `string`
- `value` : `string`

ApplicationSourceKustomize

- `apiVersions` : `[]string`
- `commonAnnotations` : `map[string]string`
- `commonAnnotationsEnvsubst` : `boolean`

- `commonLabels` : `map[string]string`
- `components` : `[]string`
- `forceCommonAnnotations` : `boolean`
- `forceCommonLabels` : `boolean`
- `images` : `[]string`
- `kubeVersion` : `string`
- `labelWithoutSelector` : `boolean`
- `namePrefix` : `string`
- `nameSuffix` : `string`
- `namespace` : `string`
- `patches` : `[]KustomizePatch`
- `replicas` : `[]KustomizeReplica`
- `version` : `string`

KustomizePatch

- `options` : `map[string]boolean`
- `patch` : `string`
- `path` : `string`
- `target` : KustomizeSelector

KustomizeSelector

- `annotationSelector` : `string`
- `group` : `string`
- `kind` : `string`
- `labelSelector` : `string`
- `name` : `string`

- `namespace` : `string`
- `version` : `string`

KustomizeReplica

- `count` : `string`
- `name` : `string`

ApplicationSourcePlugin

- `env` : `[]EnvEntry`
- `name` : `string`
- `parameters` : `[]ApplicationSourcePluginParameter`

EnvEntry

- `name` : `string`
- `value` : `string`

ApplicationSourcePluginParameter

- `OptionalArray` : OptionalArray
- `OptionalMap` : OptionalMap
- `name` : `string`
- `string` : `string`

OptionalArray

- `array` : `[]string`

OptionalMap

- `map` : `map[string]string`

SyncStrategy

- `apply` : SyncStrategyApply
- `hook` : SyncStrategyHook

SyncStrategyApply

- `force` : `boolean`

SyncStrategyHook

- `force` : `boolean`

ApplicationSpec

- `destination` : ApplicationDestination
- `ignoreDifferences` : `[]ResourceIgnoreDifferences`
- `info` : `[]Info`
- `project` : `string`
- `revisionHistoryLimit` : `integer`
- `source` : ApplicationSource
- `sourceHydrator` : SourceHydrator

- `sources` : `[]ApplicationSource`
- `syncPolicy` : SyncPolicy

ApplicationDestination

- `name` : `string`
- `namespace` : `string`
- `server` : `string`

ResourceIgnoreDifferences

- `group` : `string`
- `jqPathExpressions` : `[]string`
- `jsonPointers` : `[]string`
- `kind` : `string`
- `managedFieldsManagers` : `[]string`
- `name` : `string`
- `namespace` : `string`

SourceHydrator

- `drySource` : DrySource
- `hydrateTo` : HydrateTo
- `syncSource` : SyncSource

DrySource

- `path` : `string`

- `repoURL` : `string`
- `targetRevision` : `string`

HydrateTo

- `targetBranch` : `string`

SyncSource

- `path` : `string`
- `targetBranch` : `string`

SyncPolicy

- `automated` : SyncPolicyAutomated
- `managedNamespaceMetadata` : ManagedNamespaceMetadata
- `retry` : RetryStrategy
- `syncOptions` : `[]string`

SyncPolicyAutomated

- `allowEmpty` : `boolean`
- `prune` : `boolean`
- `selfHeal` : `boolean`

ManagedNamespaceMetadata

- `annotations` : `map[string]string`

- `labels` : `map[string]string`

ApplicationStatus

- `conditions` : `[]ApplicationCondition`
- `controllerNamespace` : `string`
- `health` : HealthStatus
- `history` : `[]RevisionHistory`
- `observedAt` : `string`
- `operationState` : OperationState
- `reconciledAt` : `string`
- `resourceHealthSource` : `string`
- `resources` : `[]ResourceStatus`
- `sourceHydrator` : SourceHydratorStatus
- `sourceType` : `string`
- `sourceTypes` : `[]string`
- `summary` : ApplicationSummary
- `sync` : SyncStatus

ApplicationCondition

- `lastTransitionTime` : `string`
- `message` : `string`
- `type` : `string`

HealthStatus

- `lastTransitionTime` : `string`

- `message` : `string`
- `status` : `string`

RevisionHistory

- `deployStartedAt` : `string`
- `deployedAt` : `string`
- `id` : `integer`
- `initiatedBy` : OperationInitiator
- `revision` : `string`
- `revisions` : `[]string`
- `source` : ApplicationSource
- `sources` : `[]ApplicationSource`

OperationState

- `finishedAt` : `string`
- `message` : `string`
- `operation` : Operation
- `phase` : `string`
- `retryCount` : `integer`
- `startedAt` : `string`
- `syncResult` : SyncOperationResult

SyncOperationResult

- `managedNamespaceMetadata` : ManagedNamespaceMetadata
- `resources` : `[]ResourceResult`

- `revision` : `string`
- `revisions` : `[]string`
- `source` : ApplicationSource
- `sources` : `[]ApplicationSource`

ResourceResult

- `group` : `string`
- `hookPhase` : `string`
- `hookType` : `string`
- `kind` : `string`
- `message` : `string`
- `name` : `string`
- `namespace` : `string`
- `status` : `string`
- `syncPhase` : `string`
- `version` : `string`

ResourceStatus

- `group` : `string`
- `health` : HealthStatus
- `hook` : `boolean`
- `kind` : `string`
- `name` : `string`
- `namespace` : `string`
- `requiresDeletionConfirmation` : `boolean`
- `requiresPruning` : `boolean`

- `status` : `string`
- `syncWave` : `integer`
- `version` : `string`

SourceHydratorStatus

- `currentOperation` : HydrateOperation
- `lastSuccessfulOperation` : SuccessfulHydrateOperation

HydrateOperation

- `drySHA` : `string`
- `finishedAt` : `string`
- `hydratedSHA` : `string`
- `message` : `string`
- `phase` : `string`
- `sourceHydrator` : SourceHydrator
- `startedAt` : `string`

SuccessfulHydrateOperation

- `drySHA` : `string`
- `hydratedSHA` : `string`
- `sourceHydrator` : SourceHydrator

ApplicationSummary

- `externalURLs` : `[]string`

- `images` : `[]string`

SyncStatus

- `comparedTo` : ComparedTo
- `revision` : `string`
- `revisions` : `[]string`
- `status` : `string`

ComparedTo

- `destination` : ApplicationDestination
- `ignoreDifferences` : `[]ResourceIgnoreDifferences`
- `source` : ApplicationSource
- `sources` : `[]ApplicationSource`

ListMeta

ListMeta describes metadata that synthetic resources must have, including lists and various status objects. A resource may have only one of {ObjectMeta, ListMeta}.

- `continue` : `string`
  continue may be set if the user set a limit on the number of items returned, and indicates that the server has more data available. The value is opaque and may be used to issue another request to the endpoint that served this list to retrieve the next set of available objects. Continuing a consistent list may not be possible if the server configuration has changed or more than a few minutes have passed. The resourceVersion field returned when using this continue value will be identical to the value in the first response, unless you have received this token from an error message.
- `remainingItemCount` : `integer`

remainingItemCount is the number of subsequent items in the list which are not included in this list response. If the list request contained label or field selectors, then the number of remaining items is unknown and the field will be left unset and omitted during serialization. If the list is complete (either because it is not chunking or because this is the last chunk), then there are no more remaining items and this field will be left unset and omitted during serialization. Servers older than v1.15 do not set this field. The intended use of the remainingItemCount is *estimating* the size of a collection. Clients should not rely on the remainingItemCount to be set or to be exact.

- `resourceVersion`: `string`
  String that identifies the server's internal version of this object that can be used by clients to determine when objects have changed. Value must be treated as opaque by clients and passed unmodified back to the server. Populated by the system. Read-only. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
- `selfLink`: `string`
  Deprecated: selfLink is a legacy read-only field that is no longer populated by the system.

ApplicationSet

/gitops/v1/project/{project}/applicationsets

`get` List applicationsets in a project

参数

- `Authorization` (*in header*): `string`
  Given Bearer token will use this as authorization for the API
- `project` (*in path*): `string` required
  project name
- `filter` (*in query*): `string`
  list filter
- `limit` (*in query*): `number`
  list limit
- `continue` (*in query*): `string`
  list continue

响应

- `200` ApplicationSetList: success

`post` Create an applicationset in a project

参数

- `Authorization` (*in header*): `string`
  Given Bearer token will use this as authorization for the API
- `project` (*in path*): `string` required
  project name

请求体

ApplicationSet required

响应

- `201` ApplicationSet: success

/gitops/v1/project/{project}/applicationsets/{name}

`get` Get an applicationset in a project

参数

- `Authorization` (*in header*): `string`
  Given Bearer token will use this as authorization for the API
- `project` (*in path*): `string` required
  project name
- `name` (*in path*): `string` required
  app name

响应

- `200` ApplicationSet: success

`put` Update an applicationset in a project

参数

- `Authorization` (*in header*): `string`
  Given Bearer token will use this as authorization for the API
- `project` (*in path*): `string` required
  project name
- `name` (*in path*): `string` required
  app name

请求体

ApplicationSet required

响应

- `200` ApplicationSet: success

`delete` Delete an applicationset in a project

参数

- `Authorization` (*in header*): `string`
  Given Bearer token will use this as authorization for the API
- `project` (*in path*): `string` required
  project name
- `name` (*in path*): `string` required
  app name

响应

- `204` : success

/gitops/v1/project/{project}/applicationsets/{name}/applications

`get` List applications in an applicationset

参数

- `Authorization` (*in header*): `string`
  Given Bearer token will use this as authorization for the API
- `project` (*in path*): `string` required
  project name
- `name` (*in path*): `string` required
  app name

响应

- `200` ApplicationList: success

/gitops/v1/project/{project}/applicationsets/{name}/applications/{app_name}

`get` Get an application in an applicationset

参数

- `Authorization` (*in header*): `string`
  Given Bearer token will use this as authorization for the API
- `project` (*in path*): `string` required
  project name
- `name` (*in path*): `string` required
  app name
- `app_name` (*in path*): `string` required
  app name

响应

- `200` Application: success

`delete` Delete an application in an applicationset

参数

- `Authorization` (*in header*): `string`
  Given Bearer token will use this as authorization for the API
- `project` (*in path*): `string` required
  project name
- `name` (*in path*): `string` required
  app name
- `app_name` (*in path*): `string` required
  app name

响应

- `204` : success

/gitops/v1/project/{project}/applicationsets/{name}/applications/{app_name}/sync

`post` Sync resources of application in an applicationset

参数

- `Authorization` (*in header*): `string`
  Given Bearer token will use this as authorization for the API
- `project` (*in path*): `string` required
  project name
- `name` (*in path*): `string` required
  app name

- `app_name` (*in path*): `string` required
  app name

响应

- `204` : success

/gitops/v1/project/{project}/applicationsets/{name}/events

`get` Query events of an applicationset and its applications

参数

- `Authorization` (*in header*): `string`
  Given Bearer token will use this as authorization for the API
- `project` (*in path*): `string` required
  project name
- `name` (*in path*): `string` required
  app name
- `start_time` (*in query*): `string`
  start time
- `end_time` (*in query*): `string`
  end time
- `reason` (*in query*): `string`
  reason
- `kind` (*in query*): `string`
  kind
- `name` (*in query*): `string`
  name
- `event_type` (*in query*): `string`

event type

- `pageno` (*in query*): `integer`
  page number
- `size` (*in query*): `size`
  size

响应

- `200` : success

/gitops/v1/project/{project}/applicationsets/{name}/prometheusrules

`get` List prometheus rules of an applicationset

参数

- `Authorization` (*in header*): `string`
  Given Bearer token will use this as authorization for the API
- `project` (*in path*): `string` required
  project name
- `name` (*in path*): `string` required
  app name
- `keyword` (*in query*): `string`
  name filter

响应

- `200` PrometheusList: success

`post` Create a prometheus rule for an applicationset

参数

- `Authorization` (*in header*): `string`
  Given Bearer token will use this as authorization for the API
- `project` (*in path*): `string` required
  project name
- `name` (*in path*): `string` required
  app name

响应

- `201` PrometheusRule: success

/gitops/v1/project/{project}/applicationsets/{name}/prometheusrules/{rule_name}

`get` Get a prometheus rule of an applicationset

参数

- `Authorization` (*in header*): `string`
  Given Bearer token will use this as authorization for the API
- `project` (*in path*): `string` required
  project name
- `name` (*in path*): `string` required
  app name
- `rule_name` (*in path*): `string` required
  rule name

响应

- `200` PrometheusRule: success

`put` Update a prometheus rule of an applicationset

参数

- `Authorization` (*in header*): `string`
  Given Bearer token will use this as authorization for the API
- `project` (*in path*): `string` required
  project name
- `name` (*in path*): `string` required
  app name
- `rule_name` (*in path*): `string` required
  rule name

响应

- `200` PrometheusRule: success

`delete` Delete a prometheus rule of an applicationset

参数

- `Authorization` (*in header*): `string`
  Given Bearer token will use this as authorization for the API
- `project` (*in path*): `string` required
  project name
- `name` (*in path*): `string` required
  app name
- `rule_name` (*in path*): `string` required
  rule name

响应

- `204` : success

`patch` Patch a prometheus rule of an applicationset

参数

- `Authorization` (*in header*): `string`
  Given Bearer token will use this as authorization for the API
- `project` (*in path*): `string` required
  project name
- `name` (*in path*): `string` required
  app name
- `rule_name` (*in path*): `string` required
  rule name

响应

- `200` PrometheusRule: success

/gitops/v1/project/{project}/applicationsets/{name}/refresh

`post` Refresh all applications in an applicationset

参数

- `Authorization` (*in header*): `string`
  Given Bearer token will use this as authorization for the API
- `project` (*in path*): `string` required
  project name
- `name` (*in path*): `string` required
  app name

响应

- `204` : success

/gitops/v1/project/{project}/applicationsets/{name}/sync

`post` Sync all applications in an applicationset

参数

- `Authorization` (*in header*): `string`
  Given Bearer token will use this as authorization for the API
- `project` (*in path*): `string` required
  project name
- `name` (*in path*): `string` required
  app name

响应

- `204` : success

/gitops/v1/project/{project}/applicationsets/{name}/topology

`post` Get (sub)topology of an applicationset

参数

- `Authorization` (*in header*): `string`
  Given Bearer token will use this as authorization for the API
- `project` (*in path*): `string` required
  project name
- `name` (*in path*): `string` required
  app name

请求体

Topology required

响应

- `204` ApplicationSetTopology: success

ApplicationSetList

- `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 ↗
- `items`: `[]ApplicationSet`
- `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`: ListMeta
  ListMeta describes metadata that synthetic resources must have, including lists and various status objects. A resource may have only one of {ObjectMeta, ListMeta}.

ApplicationSet

- `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.
- `spec` : ApplicationSetSpec
- `status` : ApplicationSetStatus

ApplicationSetSpec

- `applyNestedSelectors` : `boolean`
- `generators` : `[]ApplicationSetGenerator`
- `goTemplate` : `boolean`
- `goTemplateOptions` : `[]string`
- `ignoreApplicationDifferences` : `[]ApplicationSetResourceIgnoreDifferences`
- `preservedFields` : ApplicationPreservedFields
- `strategy` : ApplicationSetStrategy
- `syncPolicy` : ApplicationSetSyncPolicy
- `template` : ApplicationSetTemplate
- `templatePatch` : `string`

ApplicationSetGenerator

- `clusterDecisionResource` : DuckTypeGenerator
- `clusters` : ClusterGenerator
- `git` : GitGenerator
- `list` : ListGenerator

- `matrix` : MatrixGenerator
- `merge` : MergeGenerator
- `plugin` : PluginGenerator
- `pullRequest` : PullRequestGenerator
- `scmProvider` : SCMProviderGenerator
- `selector` : LabelSelector
  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.

DuckTypeGenerator

- `configMapRef` : `string`
- `labelSelector` : LabelSelector
  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.
- `name` : `string`
- `requeueAfterSeconds` : `integer`
- `template` : ApplicationSetTemplate
- `values` : `map[string]string`

LabelSelector

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.

- `matchExpressions` : `[]LabelSelectorRequirement`
  matchExpressions is a list of label selector requirements. The requirements are ANDed.
- `matchLabels` : `map[string]string`

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.

LabelSelectorRequirement

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

- `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` : `[]string`
  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.

ApplicationSetTemplate

- `metadata` : ApplicationSetTemplateMeta
- `spec` : ApplicationSpec

ApplicationSetTemplateMeta

- `annotations` : `map[string]string`
- `finalizers` : `[]string`
- `labels` : `map[string]string`
- `name` : `string`
- `namespace` : `string`

ApplicationSpec

- `destination` : ApplicationDestination
- `ignoreDifferences` : `[]ResourceIgnoreDifferences`
- `info` : `[]Info`
- `project` : `string`
- `revisionHistoryLimit` : `integer`
- `source` : ApplicationSource
- `sourceHydrator` : SourceHydrator
- `sources` : `[]ApplicationSource`
- `syncPolicy` : SyncPolicy

ApplicationDestination

- `name` : `string`
- `namespace` : `string`
- `server` : `string`

ResourceIgnoreDifferences

- `group` : `string`
- `jqPathExpressions` : `[]string`
- `jsonPointers` : `[]string`
- `kind` : `string`
- `managedFieldsManagers` : `[]string`
- `name` : `string`
- `namespace` : `string`

Info

- `name` : `string`
- `value` : `string`

ApplicationSource

- `chart` : `string`
- `directory` : ApplicationSourceDirectory
- `helm` : ApplicationSourceHelm
- `kustomize` : ApplicationSourceKustomize
- `name` : `string`
- `path` : `string`
- `plugin` : ApplicationSourcePlugin
- `ref` : `string`
- `repoURL` : `string`
- `targetRevision` : `string`

ApplicationSourceDirectory

- `exclude` : `string`
- `include` : `string`
- `jsonnet` : ApplicationSourceJsonnet
- `recurse` : `boolean`

ApplicationSourceJsonnet

- `extVars` : `[]JsonnetVar`

- `libs` : `[]string`
- `tlas` : `[]JsonnetVar`

JsonnetVar

- `code` : `boolean`
- `name` : `string`
- `value` : `string`

ApplicationSourceHelm

- `apiVersions` : `[]string`
- `fileParameters` : `[]HelmFileParameter`
- `ignoreMissingValueFiles` : `boolean`
- `kubeVersion` : `string`
- `namespace` : `string`
- `parameters` : `[]HelmParameter`
- `passCredentials` : `boolean`
- `releaseName` : `string`
- `skipCrds` : `boolean`
- `skipSchemaValidation` : `boolean`
- `skipTests` : `boolean`
- `valueFiles` : `[]string`
- `values` : `string`
- `valuesObject` : `string`
- `version` : `string`

HelmFileParameter

- `name` : `string`
- `path` : `string`

HelmParameter

- `forceString` : `boolean`
- `name` : `string`
- `value` : `string`

ApplicationSourceKustomize

- `apiVersions` : `[]string`
- `commonAnnotations` : `map[string]string`
- `commonAnnotationsEnvsubst` : `boolean`
- `commonLabels` : `map[string]string`
- `components` : `[]string`
- `forceCommonAnnotations` : `boolean`
- `forceCommonLabels` : `boolean`
- `images` : `[]string`
- `kubeVersion` : `string`
- `labelWithoutSelector` : `boolean`
- `namePrefix` : `string`
- `nameSuffix` : `string`
- `namespace` : `string`
- `patches` : `[]KustomizePatch`
- `replicas` : `[]KustomizeReplica`
- `version` : `string`

KustomizePatch

- `options` : `map[string]boolean`
- `patch` : `string`
- `path` : `string`
- `target` : KustomizeSelector

KustomizeSelector

- `annotationSelector` : `string`
- `group` : `string`
- `kind` : `string`
- `labelSelector` : `string`
- `name` : `string`
- `namespace` : `string`
- `version` : `string`

KustomizeReplica

- `count` : `string`
- `name` : `string`

ApplicationSourcePlugin

- `env` : `[]EnvEntry`
- `name` : `string`
- `parameters` : `[]ApplicationSourcePluginParameter`

EnvEntry

- `name` : `string`
- `value` : `string`

ApplicationSourcePluginParameter

- `OptionalArray` : OptionalArray
- `OptionalMap` : OptionalMap
- `name` : `string`
- `string` : `string`

OptionalArray

- `array` : `[]string`

OptionalMap

- `map` : `map[string]string`

SourceHydrator

- `drySource` : DrySource
- `hydrateTo` : HydrateTo
- `syncSource` : SyncSource

DrySource

- `path` : `string`

- `repoURL` : `string`
- `targetRevision` : `string`

HydrateTo

- `targetBranch` : `string`

SyncSource

- `path` : `string`
- `targetBranch` : `string`

SyncPolicy

- `automated` : SyncPolicyAutomated
- `managedNamespaceMetadata` : ManagedNamespaceMetadata
- `retry` : RetryStrategy
- `syncOptions` : `[]string`

SyncPolicyAutomated

- `allowEmpty` : `boolean`
- `prune` : `boolean`
- `selfHeal` : `boolean`

ManagedNamespaceMetadata

- `annotations` : `map[string]string`

- `labels` : `map[string]string`

RetryStrategy

- `backoff` : Backoff
- `limit` : `integer`

Backoff

- `duration` : `string`
- `factor` : `integer`
- `maxDuration` : `string`

ClusterGenerator

- `flatList` : `boolean`
- `selector` : LabelSelector
  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.
- `template` : ApplicationSetTemplate
- `values` : `map[string]string`

GitGenerator

- `directories` : `[]GitDirectoryGeneratorItem`
- `files` : `[]GitFileGeneratorItem`
- `pathParamPrefix` : `string`
- `repoURL` : `string`

- `requeueAfterSeconds` : `integer`
- `revision` : `string`
- `template` : ApplicationSetTemplate
- `values` : `map[string]string`

GitDirectoryGeneratorItem

- `exclude` : `boolean`
- `path` : `string`

GitFileGeneratorItem

- `path` : `string`

ListGenerator

- `elements` : `[]JSON`
- `elementsYaml` : `string`
- `template` : ApplicationSetTemplate

JSON

MatrixGenerator

- `generators` : `[]ApplicationSetNestedGenerator`
- `template` : ApplicationSetTemplate

ApplicationSetNestedGenerator

- `clusterDecisionResource` : DuckTypeGenerator
- `clusters` : ClusterGenerator
- `git` : GitGenerator
- `list` : ListGenerator
- `matrix` : `string`
- `merge` : `string`
- `plugin` : PluginGenerator
- `pullRequest` : PullRequestGenerator
- `scmProvider` : SCMProviderGenerator
- `selector` : LabelSelector
  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.

PluginGenerator

- `configMapRef` : PluginConfigMapRef
- `input` : PluginInput
- `requeueAfterSeconds` : `integer`
- `template` : ApplicationSetTemplate
- `values` : `map[string]string`

PluginConfigMapRef

- `name` : `string`

PluginInput

- `parameters` : `map[string]JSON`

PullRequestGenerator

- `azuredevops` : PullRequestGeneratorAzureDevOps
- `bitbucket` : PullRequestGeneratorBitbucket
- `bitbucketServer` : PullRequestGeneratorBitbucketServer
- `filters` : `[]PullRequestGeneratorFilter`
- `gitea` : PullRequestGeneratorGitea
- `github` : PullRequestGeneratorGithub
- `gitlab` : PullRequestGeneratorGitLab
- `requeueAfterSeconds` : `integer`
- `template` : ApplicationSetTemplate

PullRequestGeneratorAzureDevOps

- `api` : `string`
- `labels` : `[]string`
- `organization` : `string`
- `project` : `string`
- `repo` : `string`
- `tokenRef` : SecretRef

SecretRef

- `key` : `string`

- `secretName` : `string`

PullRequestGeneratorBitbucket

- `api` : `string`
- `basicAuth` : BasicAuthBitbucketServer
- `bearerToken` : BearerTokenBitbucketCloud
- `owner` : `string`
- `repo` : `string`

BasicAuthBitbucketServer

- `passwordRef` : SecretRef
- `username` : `string`

BearerTokenBitbucketCloud

- `tokenRef` : SecretRef

PullRequestGeneratorBitbucketServer

- `api` : `string`
- `basicAuth` : BasicAuthBitbucketServer
- `bearerToken` : BearerTokenBitbucket
- `caRef` : ConfigMapKeyRef
- `insecure` : `boolean`
- `project` : `string`
- `repo` : `string`

BearerTokenBitbucket

- `tokenRef` : SecretRef

ConfigMapKeyRef

- `configMapName` : `string`
- `key` : `string`

PullRequestGeneratorFilter

- `branchMatch` : `string`
- `targetBranchMatch` : `string`

PullRequestGeneratorGitea

- `api` : `string`
- `insecure` : `boolean`
- `owner` : `string`
- `repo` : `string`
- `tokenRef` : SecretRef

PullRequestGeneratorGithub

- `api` : `string`
- `appSecretName` : `string`
- `labels` : `[]string`
- `owner` : `string`

- `repo` : `string`
- `tokenRef` : SecretRef

PullRequestGeneratorGitLab

- `api` : `string`
- `caRef` : ConfigMapKeyRef
- `insecure` : `boolean`
- `labels` : `[]string`
- `project` : `string`
- `pullRequestState` : `string`
- `tokenRef` : SecretRef

SCMProviderGenerator

- `awsCodeCommit` : SCMProviderGeneratorAWSCodeCommit
- `azureDevOps` : SCMProviderGeneratorAzureDevOps
- `bitbucket` : SCMProviderGeneratorBitbucket
- `bitbucketServer` : SCMProviderGeneratorBitbucketServer
- `cloneProtocol` : `string`
- `filters` : `[]SCMProviderGeneratorFilter`
- `gitea` : SCMProviderGeneratorGitea
- `github` : SCMProviderGeneratorGithub
- `gitlab` : SCMProviderGeneratorGitlab
- `requeueAfterSeconds` : `integer`
- `template` : ApplicationSetTemplate
- `values` : `map[string]string`

SCMProviderGeneratorAWSCodeCommit

- `allBranches` : `boolean`
- `region` : `string`
- `role` : `string`
- `tagFilters` : `[]TagFilter`

TagFilter

- `key` : `string`
- `value` : `string`

SCMProviderGeneratorAzureDevOps

- `accessTokenRef` : SecretRef
- `allBranches` : `boolean`
- `api` : `string`
- `organization` : `string`
- `teamProject` : `string`

SCMProviderGeneratorBitbucket

- `allBranches` : `boolean`
- `appPasswordRef` : SecretRef
- `owner` : `string`
- `user` : `string`

SCMProviderGeneratorBitbucketServer

- `allBranches` : `boolean`
- `api` : `string`
- `basicAuth` : BasicAuthBitbucketServer
- `bearerToken` : BearerTokenBitbucket
- `caRef` : ConfigMapKeyRef
- `insecure` : `boolean`
- `project` : `string`

SCMProviderGeneratorFilter

- `branchMatch` : `string`
- `labelMatch` : `string`
- `pathsDoNotExist` : `[]string`
- `pathsExist` : `[]string`
- `repositoryMatch` : `string`

SCMProviderGeneratorGitea

- `allBranches` : `boolean`
- `api` : `string`
- `insecure` : `boolean`
- `owner` : `string`
- `tokenRef` : SecretRef

SCMProviderGeneratorGithub

- `allBranches` : `boolean`
- `api` : `string`

- `appSecretName` : `string`
- `organization` : `string`
- `tokenRef` : SecretRef

SCMProviderGeneratorGitlab

- `allBranches` : `boolean`
- `api` : `string`
- `caRef` : ConfigMapKeyRef
- `group` : `string`
- `includeSharedProjects` : `boolean`
- `includeSubgroups` : `boolean`
- `insecure` : `boolean`
- `tokenRef` : SecretRef
- `topic` : `string`

MergeGenerator

- `generators` : `[]ApplicationSetNestedGenerator`
- `mergeKeys` : `[]string`
- `template` : ApplicationSetTemplate

ApplicationSetResourceIgnoreDifferences

- `jqPathExpressions` : `[]string`
- `jsonPointers` : `[]string`
- `name` : `string`

ApplicationPreservedFields

- `annotations` : `[]string`
- `labels` : `[]string`

ApplicationSetStrategy

- `rollingSync` : ApplicationSetRolloutStrategy
- `type` : `string`

ApplicationSetRolloutStrategy

- `steps` : `[]ApplicationSetRolloutStep`

ApplicationSetRolloutStep

- `matchExpressions` : `[]ApplicationMatchExpression`
- `maxUpdate` : `string`

ApplicationMatchExpression

- `key` : `string`
- `operator` : `string`
- `values` : `[]string`

ApplicationSetSyncPolicy

- `applicationsSync` : `string`
- `preserveResourcesOnDeletion` : `boolean`

ApplicationSetStatus

- `applicationStatus` : `[]ApplicationSetApplicationStatus`
- `conditions` : `[]ApplicationSetCondition`
- `resources` : `[]ResourceStatus`

ApplicationSetApplicationStatus

- `application` : `string`
- `lastTransitionTime` : `string`
- `message` : `string`
- `status` : `string`
- `step` : `string`
- `targetRevisions` : `[]string`

ApplicationSetCondition

- `lastTransitionTime` : `string`
- `message` : `string`
- `reason` : `string`
- `status` : `string`
- `type` : `string`

ResourceStatus

- `group` : `string`
- `health` : HealthStatus
- `hook` : `boolean`

- `kind` : `string`
- `name` : `string`
- `namespace` : `string`
- `requiresDeletionConfirmation` : `boolean`
- `requiresPruning` : `boolean`
- `status` : `string`
- `syncWave` : `integer`
- `version` : `string`

HealthStatus

- `lastTransitionTime` : `string`
- `message` : `string`
- `status` : `string`

ListMeta

ListMeta describes metadata that synthetic resources must have, including lists and various status objects. A resource may have only one of {ObjectMeta, ListMeta}.

- `continue` : `string`
  continue may be set if the user set a limit on the number of items returned, and indicates that the server has more data available. The value is opaque and may be used to issue another request to the endpoint that served this list to retrieve the next set of available objects. Continuing a consistent list may not be possible if the server configuration has changed or more than a few minutes have passed. The resourceVersion field returned when using this continue value will be identical to the value in the first response, unless you have received this token from an error message.
- `remainingItemCount` : `integer`
  remainingItemCount is the number of subsequent items in the list which are not included in this list response. If the list request contained label or field selectors, then the number of

remaining items is unknown and the field will be left unset and omitted during serialization. If the list is complete (either because it is not chunking or because this is the last chunk), then there are no more remaining items and this field will be left unset and omitted during serialization. Servers older than v1.15 do not set this field. The intended use of the remainingItemCount is *estimating* the size of a collection. Clients should not rely on the remainingItemCount to be set or to be exact.

- `resourceVersion`: `string`
  String that identifies the server's internal version of this object that can be used by clients to determine when objects have changed. Value must be treated as opaque by clients and passed unmodified back to the server. Populated by the system. Read-only. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
- `selfLink`: `string`
  Deprecated: selfLink is a legacy read-only field that is no longer populated by the system.

ApplicationList

- `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 ↗
- `items`: `[]Application`
- `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`: ListMeta
  ListMeta describes metadata that synthetic resources must have, including lists and various status objects. A resource may have only one of {ObjectMeta, ListMeta}.

Application

- `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.
- `operation` : Operation
- `spec` : ApplicationSpec
- `status` : ApplicationStatus

Operation

- `info` : `[]Info`
- `initiatedBy` : OperationInitiator
- `retry` : RetryStrategy
- `sync` : SyncOperation

OperationInitiator

- `automated` : `boolean`
- `username` : `string`

SyncOperation

- `autoHealAttemptsCount` : `integer`
- `dryRun` : `boolean`
- `manifests` : `[]string`
- `prune` : `boolean`
- `resources` : `[]SyncOperationResource`
- `revision` : `string`
- `revisions` : `[]string`
- `source` : ApplicationSource
- `sources` : `[]ApplicationSource`
- `syncOptions` : `[]string`
- `syncStrategy` : SyncStrategy

SyncOperationResource

- `group` : `string`
- `kind` : `string`
- `name` : `string`
- `namespace` : `string`

SyncStrategy

- `apply` : SyncStrategyApply
- `hook` : SyncStrategyHook

SyncStrategyApply

- `force` : `boolean`

SyncStrategyHook

- `force` : `boolean`

ApplicationStatus

- `conditions` : `[]ApplicationCondition`
- `controllerNamespace` : `string`
- `health` : HealthStatus
- `history` : `[]RevisionHistory`
- `observedAt` : `string`
- `operationState` : OperationState
- `reconciledAt` : `string`
- `resourceHealthSource` : `string`
- `resources` : `[]ResourceStatus`
- `sourceHydrator` : SourceHydratorStatus
- `sourceType` : `string`
- `sourceTypes` : `[]string`
- `summary` : ApplicationSummary
- `sync` : SyncStatus

ApplicationCondition

- `lastTransitionTime` : `string`
- `message` : `string`
- `type` : `string`

RevisionHistory

- `deployStartedAt` : `string`
- `deployedAt` : `string`
- `id` : `integer`
- `initiatedBy` : OperationInitiator
- `revision` : `string`
- `revisions` : `[]string`
- `source` : ApplicationSource
- `sources` : `[]ApplicationSource`

OperationState

- `finishedAt` : `string`
- `message` : `string`
- `operation` : Operation
- `phase` : `string`
- `retryCount` : `integer`
- `startedAt` : `string`
- `syncResult` : SyncOperationResult

SyncOperationResult

- `managedNamespaceMetadata` : ManagedNamespaceMetadata
- `resources` : `[]ResourceResult`
- `revision` : `string`
- `revisions` : `[]string`
- `source` : ApplicationSource

- `sources` : `[]ApplicationSource`

ResourceResult

- `group` : `string`
- `hookPhase` : `string`
- `hookType` : `string`
- `kind` : `string`
- `message` : `string`
- `name` : `string`
- `namespace` : `string`
- `status` : `string`
- `syncPhase` : `string`
- `version` : `string`

SourceHydratorStatus

- `currentOperation` : HydrateOperation
- `lastSuccessfulOperation` : SuccessfulHydrateOperation

HydrateOperation

- `drySHA` : `string`
- `finishedAt` : `string`
- `hydratedSHA` : `string`
- `message` : `string`
- `phase` : `string`
- `sourceHydrator` : SourceHydrator

- `startedAt` : `string`

SuccessfulHydrateOperation

- `drySHA` : `string`
- `hydratedSHA` : `string`
- `sourceHydrator` : SourceHydrator

ApplicationSummary

- `externalURLs` : `[]string`
- `images` : `[]string`

SyncStatus

- `comparedTo` : ComparedTo
- `revision` : `string`
- `revisions` : `[]string`
- `status` : `string`

ComparedTo

- `destination` : ApplicationDestination
- `ignoreDifferences` : `[]ResourceIgnoreDifferences`
- `source` : ApplicationSource
- `sources` : `[]ApplicationSource`

PrometheusList

- `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 ↗
- `items` : `[]Prometheus`
- `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` : ListMeta
  ListMeta describes metadata that synthetic resources must have, including lists and various status objects. A resource may have only one of {ObjectMeta, ListMeta}.

Prometheus

- `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.
- `spec` : PrometheusSpec
- `status` : PrometheusStatus

PrometheusSpec

- `additionalAlertManagerConfigs` : SecretKeySelector

  SecretKeySelector selects a key of a Secret.

- `additionalAlertRelabelConfigs` : SecretKeySelector

  SecretKeySelector selects a key of a Secret.

- `additionalArgs` : `[]Argument`

- `additionalScrapeConfigs` : SecretKeySelector

  SecretKeySelector selects a key of a Secret.

- `affinity` : Affinity

  Affinity is a group of affinity scheduling rules.

- `alerting` : AlertingSpec

- `allowOverlappingBlocks` : `boolean`

- `apiserverConfig` : APIServerConfig

- `arbitraryFSAccessThroughSMs` : ArbitraryFSAccessThroughSMsConfig

- `automountServiceAccountToken` : `boolean`

- `baseImage` : `string`

- `bodySizeLimit` : `string`

- `configMaps` : `[]string`

- `containers` : `[]Container`

- `disableCompaction` : `boolean`

- `enableAdminAPI` : `boolean`

- `enableFeatures` : `[]string`

- `enableRemoteWriteReceiver` : `boolean`

- `enforcedBodySizeLimit` : `string`

- `enforcedKeepDroppedTargets` : `integer`

- `enforcedLabelLimit` : `integer`

- `enforcedLabelNameLengthLimit` : `integer`

- `enforcedLabelValueLengthLimit` : `integer`

- `enforcedNamespaceLabel` : `string`
- `enforcedSampleLimit` : `integer`
- `enforcedTargetLimit` : `integer`
- `evaluationInterval` : `string`
- `excludedFromEnforcement` : `[]ObjectReference`
- `exemplars` : Exemplars
- `externalLabels` : `map[string]string`
- `externalUrl` : `string`
- `hostAliases` : `[]HostAlias`
- `hostNetwork` : `boolean`
- `ignoreNamespaceSelectors` : `boolean`
- `image` : `string`
- `imagePullPolicy` : `string`
- `imagePullSecrets` : `[]LocalObjectReference`
- `initContainers` : `[]Container`
- `keepDroppedTargets` : `integer`
- `labelLimit` : `integer`
- `labelNameLengthLimit` : `integer`
- `labelValueLengthLimit` : `integer`
- `listenLocal` : `boolean`
- `logFormat` : `string`
- `logLevel` : `string`
- `maximumStartupDurationSeconds` : `integer`
- `minReadySeconds` : `integer`
- `nodeSelector` : `map[string]string`
- `overrideHonorLabels` : `boolean`
- `overrideHonorTimestamps` : `boolean`
- `paused` : `boolean`

- `persistentVolumeClaimRetentionPolicy` : StatefulSetPersistentVolumeClaimRetentionPolicy
  StatefulSetPersistentVolumeClaimRetentionPolicy describes the policy used for PVCs created from the StatefulSet VolumeClaimTemplates.
- `podMetadata` : EmbeddedObjectMetadata
- `podMonitorNamespaceSelector` : LabelSelector
  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.
- `podMonitorSelector` : LabelSelector
  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.
- `podTargetLabels` : `[]string`
- `portName` : `string`
- `priorityClassName` : `string`
- `probeNamespaceSelector` : LabelSelector
  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.
- `probeSelector` : LabelSelector
  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.
- `prometheusExternalLabelName` : `string`
- `prometheusRulesExcludedFromEnforce` : `[]PrometheusRuleExcludeConfig`
- `query` : QuerySpec
- `queryLogFile` : `string`
- `reloadStrategy` : `string`
- `remoteRead` : `[]RemoteReadSpec`
- `remoteWrite` : `[]RemoteWriteSpec`
- `replicaExternalLabelName` : `string`

- `replicas` : `integer`
- `resources` : ResourceRequirements
  ResourceRequirements describes the compute resource requirements.
- `retention` : `string`
- `retentionSize` : `string`
- `routePrefix` : `string`
- `ruleNamespaceSelector` : LabelSelector
  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.
- `ruleSelector` : LabelSelector
  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.
- `rules` : Rules
- `sampleLimit` : `integer`
- `scrapeClasses` : `[]ScrapeClass`
- `scrapeConfigNamespaceSelector` : LabelSelector
  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.
- `scrapeConfigSelector` : LabelSelector
  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.
- `scrapeInterval` : `string`
- `scrapeProtocols` : `[]string`
- `scrapeTimeout` : `string`
- `secrets` : `[]string`
- `securityContext` : PodSecurityContext
  PodSecurityContext holds pod-level security attributes and common container settings. Some fields are also present in container.securityContext. Field values of

container.securityContext take precedence over field values of PodSecurityContext.

- `serviceAccountName` : `string`
- `serviceDiscoveryRole` : `string`
- `serviceMonitorNamespaceSelector` : LabelSelector
  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.
- `serviceMonitorSelector` : LabelSelector
  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.
- `sha` : `string`
- `shards` : `integer`
- `storage` : StorageSpec
- `tag` : `string`
- `targetLimit` : `integer`
- `thanos` : ThanosSpec
- `tolerations` : `[]Toleration`
- `topologySpreadConstraints` : `[]TopologySpreadConstraint`
- `tracingConfig` : PrometheusTracingConfig
- `tsdb` : TSDBSpec
- `version` : `string`
- `volumeMounts` : `[]VolumeMount`
- `volumes` : `[]Volume`
- `walCompression` : `boolean`
- `web` : PrometheusWebSpec

SecretKeySelector

SecretKeySelector selects a key of a Secret.

- `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

Argument

- `name` : `string`
- `value` : `string`

Affinity

Affinity is a group of affinity scheduling rules.

- `nodeAffinity` : NodeAffinity
  Node affinity is a group of node affinity scheduling rules.
- `podAffinity` : PodAffinity
  Pod affinity is a group of inter pod affinity scheduling rules.
- `podAntiAffinity` : PodAntiAffinity
  Pod anti affinity is a group of inter pod anti affinity scheduling rules.

NodeAffinity

Node affinity is a group of node affinity scheduling rules.

- `preferredDuringSchedulingIgnoredDuringExecution` : `[]PreferredSchedulingTerm`

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

- `requiredDuringSchedulingIgnoredDuringExecution` : NodeSelector
  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.

PreferredSchedulingTerm

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

- `preference` : NodeSelectorTerm
  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.
- `weight` : `integer`
  Weight associated with matching the corresponding nodeSelectorTerm, in the range 1-100.

NodeSelectorTerm

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.

- `matchExpressions` : `[]NodeSelectorRequirement`
  A list of node selector requirements by node's labels.
- `matchFields` : `[]NodeSelectorRequirement`
  A list of node selector requirements by node's fields.

NodeSelectorRequirement

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

- `key` : `string`
  The label key that the selector applies to.
- `operator` : `string`
  Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
- `values` : `[]string`
  An array of string values. If the operator 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.

NodeSelector

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.

- `nodeSelectorTerms` : `[]NodeSelectorTerm`
  Required. A list of node selector terms. The terms are ORed.

PodAffinity

Pod affinity is a group of inter pod affinity scheduling rules.

- `preferredDuringSchedulingIgnoredDuringExecution` : `[]WeightedPodAffinityTerm`
  The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through

the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

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

WeightedPodAffinityTerm

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

- `podAffinityTerm` : PodAffinityTerm
  Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key matches that of any node on which a pod of the set of pods is running
- `weight` : `integer`
  weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

PodAffinityTerm

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

- `labelSelector` : LabelSelector
  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.

- `matchLabelKeys` : `[]string`
  MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key in (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
- `mismatchLabelKeys` : `[]string`
  MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `labelSelector` as `key notin (value)` to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
- `namespaceSelector` : LabelSelector
  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.
- `namespaces` : `[]string`
  namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
- `topologyKey` : `string`
  This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

PodAntiAffinity

Pod anti affinity is a group of inter pod anti affinity scheduling rules.

- `preferredDuringSchedulingIgnoredDuringExecution` : `[]WeightedPodAffinityTerm`
  The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling anti-affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.
- `requiredDuringSchedulingIgnoredDuringExecution` : `[]PodAffinityTerm`
  If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

AlertingSpec

- `alertmanagers` : `[]AlertmanagerEndpoints`

AlertmanagerEndpoints

- `alertRelabelings` : `[]RelabelConfig`
- `apiVersion` : `string`
- `authorization` : SafeAuthorization
- `basicAuth` : BasicAuth
- `bearerTokenFile` : `string`

- `enableHttp2` : `boolean`
- `name` : `string`
- `namespace` : `string`
- `pathPrefix` : `string`
- `port` : `string`
- `relabelings` : `[]RelabelConfig`
- `scheme` : `string`
- `sigv4` : Sigv4
- `timeout` : `string`
- `tlsConfig` : TLSConfig

RelabelConfig

- `action` : `string`
- `modulus` : `integer`
- `regex` : `string`
- `replacement` : `string`
- `separator` : `string`
- `sourceLabels` : `[]string`
- `targetLabel` : `string`

SafeAuthorization

- `credentials` : SecretKeySelector
  SecretKeySelector selects a key of a Secret.
- `type` : `string`

BasicAuth

- `password` : SecretKeySelector

  SecretKeySelector selects a key of a Secret.

- `username` : SecretKeySelector

  SecretKeySelector selects a key of a Secret.

Sigv4

- `accessKey` : SecretKeySelector

  SecretKeySelector selects a key of a Secret.

- `profile` : `string`
- `region` : `string`
- `roleArn` : `string`
- `secretKey` : SecretKeySelector

  SecretKeySelector selects a key of a Secret.

TLSConfig

- `ca` : SecretOrConfigMap
- `caFile` : `string`
- `cert` : SecretOrConfigMap
- `certFile` : `string`
- `insecureSkipVerify` : `boolean`
- `keyFile` : `string`
- `keySecret` : SecretKeySelector

  SecretKeySelector selects a key of a Secret.

- `maxVersion` : `string`
- `minVersion` : `string`
- `serverName` : `string`

SecretOrConfigMap

- `configMap`: ConfigMapKeySelector
  Selects a key from a ConfigMap.
- `secret`: SecretKeySelector
  SecretKeySelector selects a key of a Secret.

ConfigMapKeySelector

Selects a key from a ConfigMap.

- `key`: `string`
  The key to select.
- `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 ConfigMap or its key must be defined

APIServerConfig

- `authorization`: Authorization
- `basicAuth`: BasicAuth
- `bearerToken`: `string`
- `bearerTokenFile`: `string`
- `host`: `string`
- `tlsConfig`: TLSConfig

Authorization

- `credentials` : SecretKeySelector
  SecretKeySelector selects a key of a Secret.
- `credentialsFile` : `string`
- `type` : `string`

ArbitraryFSAccessThroughSMsConfig

- `deny` : `boolean`

Container

A single application container that you want to run within a pod.

- `args` : `[]string`
  Arguments to the entrypoint. The container image's CMD is used if this is not provided. Variable references $(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double $$ are reduced to a single $, which allows for escaping the $(VAR_NAME) syntax: i.e. "$$(VAR_NAME)" will produce the string literal "$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell ↗
- `command` : `[]string`
  Entrypoint array. Not executed within a shell. The container image's ENTRYPOINT is used if this is not provided. Variable references $(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double $$ are reduced to a single $, which allows for escaping the $(VAR_NAME) syntax: i.e. "$$(VAR_NAME)" will produce the string literal "$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell ↗

- `env` : `[]EnvVar`
  List of environment variables to set in the container. Cannot be updated.
- `envFrom` : `[]EnvFromSource`
  List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.
- `image` : `string`
  Container image name. More info: https://kubernetes.io/docs/concepts/containers/images ↗
  This field is optional to allow higher level config management to default or override container images in workload controllers like Deployments and StatefulSets.
- `imagePullPolicy` : `string`
  Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/images#updating-images ↗
- `lifecycle` : Lifecycle
  Lifecycle describes actions that the management system should take in response to container lifecycle events. For the PostStart and PreStop lifecycle handlers, management of the container blocks until the action is complete, unless the container process fails, in which case the handler is aborted.
- `livenessProbe` : Probe
  Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
- `name` : `string`
  Name of the container specified as a DNS_LABEL. Each container in a pod must have a unique name (DNS_LABEL). Cannot be updated.
- `ports` : `[]ContainerPort`
  List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address inside a container will be accessible from the network. Modifying this array with strategic merge patch may corrupt the data. For more information See https://github.com/kubernetes/kubernetes/issues/108255 ↗. Cannot be updated.
- `readinessProbe` : Probe

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

- `resizePolicy` : `[]ContainerResizePolicy`
  Resources resize policy for the container.
- `resources` : ResourceRequirements
  ResourceRequirements describes the compute resource requirements.
- `restartPolicy` : `string`
  RestartPolicy defines the restart behavior of individual containers in a pod. This field may only be set for init containers, and the only allowed value is "Always". For non-init containers or when this field is not specified, the restart behavior is defined by the Pod's restart policy and the container type. Setting the RestartPolicy as "Always" for the init container will have the following effect: this init container will be continually restarted on exit until all regular containers have terminated. Once all regular containers have completed, all init containers with restartPolicy "Always" will be shut down. This lifecycle differs from normal init containers and is often referred to as a "sidecar" container. Although this init container still starts in the init container sequence, it does not wait for the container to complete before proceeding to the next init container. Instead, the next init container starts immediately after this init container is started, or after any startupProbe has successfully completed.
- `securityContext` : SecurityContext
  SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.
- `startupProbe` : Probe
  Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
- `stdin` : `boolean`
  Whether this container should allocate a buffer for stdin in the container runtime. If this is not set, reads from stdin in the container will always result in EOF. Default is false.
- `stdinOnce` : `boolean`
  Whether the container runtime should close the stdin channel after it has been opened by a single attach. When stdin is true the stdin stream will remain open across multiple attach sessions. If stdinOnce is set to true, stdin is opened on container start, is empty until the first client attaches to stdin, and then remains open and accepts data until the client disconnects, at which time stdin is closed and remains closed until the container is

restarted. If this flag is false, a container processes that reads from stdin will never receive an EOF. Default is false

- `terminationMessagePath` : `string`
  Optional: Path at which the file to which the container's termination message will be written is mounted into the container's filesystem. Message written is intended to be brief final status, such as an assertion failure message. Will be truncated by the node if greater than 4096 bytes. The total message length across all containers will be limited to 12kb. Defaults to /dev/termination-log. Cannot be updated.
- `terminationMessagePolicy` : `string`
  Indicate how the termination message should be populated. File will use the contents of terminationMessagePath to populate the container status message on both success and failure. FallbackToLogsOnError will use the last chunk of container log output if the termination message file is empty and the container exited with an error. The log output is limited to 2048 bytes or 80 lines, whichever is smaller. Defaults to File. Cannot be updated.
- `tty` : `boolean`
  Whether this container should allocate a TTY for itself, also requires 'stdin' to be true. Default is false.
- `volumeDevices` : `[]VolumeDevice`
  volumeDevices is the list of block devices to be used by the container.
- `volumeMounts` : `[]VolumeMount`
  Pod volumes to mount into the container's filesystem. Cannot be updated.
- `workingDir` : `string`
  Container's working directory. If not specified, the container runtime's default will be used, which might be configured in the container image. Cannot be updated.

EnvVar

EnvVar represents an environment variable present in a Container.

- `name` : `string`
  Name of the environment variable. Must be a C_IDENTIFIER.
- `value` : `string`
  Variable references $(VAR_NAME) are expanded using the previously defined environment variables in the container and any service environment variables. If a variable cannot be

resolved, the reference in the input string will be unchanged. Double $$ are reduced to a single $, which allows for escaping the $(VAR_NAME) syntax: i.e. "$$(VAR_NAME)" will produce the string literal "$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".

- `valueFrom`: EnvVarSource
  EnvVarSource represents a source for the value of an EnvVar.

EnvVarSource

EnvVarSource represents a source for the value of an EnvVar.

- `configMapKeyRef`: ConfigMapKeySelector
  Selects a key from a ConfigMap.
- `fieldRef`: ObjectFieldSelector
  ObjectFieldSelector selects an APIVersioned field of an object.
- `resourceFieldRef`: ResourceFieldSelector
  ResourceFieldSelector represents container resources (cpu, memory) and their output format
- `secretKeyRef`: SecretKeySelector
  SecretKeySelector selects a key of a Secret.

ObjectFieldSelector

ObjectFieldSelector selects an APIVersioned field of an object.

- `apiVersion`: `string`
  Version of the schema the FieldPath is written in terms of, defaults to "v1".
- `fieldPath`: `string`
  Path of the field to select in the specified API version.

ResourceFieldSelector

ResourceFieldSelector represents container resources (cpu, memory) and their output format

- `containerName` : `string`
  Container name: required for volumes, optional for env vars
- `divisor` : `string`
  Specifies the output format of the exposed resources, defaults to "1"
- `resource` : `string`
  Required: resource to select

EnvFromSource

EnvFromSource represents the source of a set of ConfigMaps

- `configMapRef` : ConfigMapEnvSource
  ConfigMapEnvSource selects a ConfigMap to populate the environment variables with.
  The contents of the target ConfigMap's Data field will represent the key-value pairs as environment variables.
- `prefix` : `string`
  An optional identifier to prepend to each key in the ConfigMap. Must be a C_IDENTIFIER.
- `secretRef` : SecretEnvSource
  SecretEnvSource selects a Secret to populate the environment variables with.
  The contents of the target Secret's Data field will represent the key-value pairs as environment variables.

ConfigMapEnvSource

ConfigMapEnvSource selects a ConfigMap to populate the environment variables with.

The contents of the target ConfigMap's Data field will represent the key-value pairs as environment variables.

- `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 ConfigMap must be defined

SecretEnvSource

SecretEnvSource selects a Secret to populate the environment variables with.

The contents of the target Secret's Data field will represent the key-value pairs as environment variables.

- `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 must be defined

Lifecycle

Lifecycle describes actions that the management system should take in response to container lifecycle events. For the PostStart and PreStop lifecycle handlers, management of the container blocks until the action is complete, unless the container process fails, in which case the handler is aborted.

- `postStart` : LifecycleHandler
  LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.
- `preStop` : LifecycleHandler
  LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.

LifecycleHandler

LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.

- `exec`: ExecAction
  ExecAction describes a "run in container" action.
- `httpGet`: HTTPGetAction
  HTTPGetAction describes an action based on HTTP Get requests.
- `sleep`: SleepAction
  SleepAction describes a "sleep" action.
- `tcpSocket`: TCPSocketAction
  TCPSocketAction describes an action based on opening a socket

ExecAction

ExecAction describes a "run in container" action.

- `command`: `[]string`
  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.

HTTPGetAction

HTTPGetAction describes an action based on HTTP Get requests.

- `host`: `string`
  Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
- `httpHeaders`: `[]HTTPHeader`
  Custom headers to set in the request. HTTP allows repeated headers.

- `path` : `string`
  Path to access on the HTTP server.
- `port` : `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.

HTTPHeader

HTTPHeader describes a custom header to be used in HTTP probes

- `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

SleepAction

SleepAction describes a "sleep" action.

- `seconds` : `integer`
  Seconds is the number of seconds to sleep.

TCPSocketAction

TCPSocketAction describes an action based on opening a socket

- `host` : `string`
  Optional: Host name to connect to, defaults to the pod IP.
- `port` : `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.

Probe

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

- `exec`: ExecAction
  ExecAction describes a "run in container" action.
- `failureThreshold`: `integer`
  Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
- `grpc`: GRPCAction
  GRPCAction specifies an action involving a GRPC service.
- `httpGet`: HTTPGetAction
  HTTPGetAction describes an action based on HTTP Get requests.
- `initialDelaySeconds`: `integer`
  Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes ↗
- `periodSeconds`: `integer`
  How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
- `successThreshold`: `integer`
  Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
- `tcpSocket`: TCPSocketAction
  TCPSocketAction describes an action based on opening a socket
- `terminationGracePeriodSeconds`: `integer`
  Optional duration in seconds the pod needs to terminate gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. If this value is nil, the

pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta field and requires enabling ProbeTerminationGracePeriod feature gate. Minimum value is 1. spec.terminationGracePeriodSeconds is used if unset.

- `timeoutSeconds` : `integer`
  Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes ↗

GRPCAction

GRPCAction specifies an action involving a GRPC service.

- `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.

ContainerPort

ContainerPort represents a network port in a single container.

- `containerPort` : `integer`
  Number of port to expose on the pod's IP address. This must be a valid port number, 0 < x < 65536.
- `hostIP` : `string`
  What host IP to bind the external port to.
- `hostPort` : `integer`
  Number of port to expose on the host. If specified, this must be a valid port number, 0 < x < 65536. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this.

- `name` : `string`
  If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services.
- `protocol` : `string`
  Protocol for port. Must be UDP, TCP, or SCTP. Defaults to "TCP".

ContainerResizePolicy

ContainerResizePolicy represents resource resize policy for the container.

- `resourceName` : `string`
  Name of the resource to which this resource resize policy applies. Supported values: cpu, memory.
- `restartPolicy` : `string`
  Restart policy to apply when specified resource is resized. If not specified, it defaults to NotRequired.

ResourceRequirements

ResourceRequirements describes the compute resource requirements.

- `claims` : `[]ResourceClaim`
  Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container.
  This is an alpha field and requires enabling the DynamicResourceAllocation feature gate.
  This field is immutable. It can only be set for containers.
- `limits` : `map[string]Quantity`
  Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗
- `requests` : `map[string]Quantity`
  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/ ↗

ResourceClaim

ResourceClaim references one entry in PodSpec.ResourceClaims.

- `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.

Quantity

- `Format` : `string`
- `d` : InfDecAmount
- `i` : Int64Amount
- `s` : `string`

InfDecAmount

- `Dec` : Dec

Dec

- `scale` : `integer`
- `unscaled` : Int

Int

- `abs` : `[]integer`
- `neg` : `boolean`

Int64Amount

- `scale` : `integer`
- `value` : `integer`

SecurityContext

SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.

- `allowPrivilegeEscalation` : `boolean`
  AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the no_new_privs flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: 1) run as Privileged 2) has CAP_SYS_ADMIN Note that this field cannot be set when spec.os.name is windows.
- `appArmorProfile` : AppArmorProfile
  AppArmorProfile defines a pod or container's AppArmor settings.
- `capabilities` : Capabilities
  Adds and removes POSIX capabilities from running containers.
- `privileged` : `boolean`
  Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.
- `procMount` : `string`

procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows.

- `readOnlyRootFilesystem` : `boolean`
  Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
- `runAsGroup` : `integer`
  The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
- `runAsNonRoot` : `boolean`
  Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
- `runAsUser` : `integer`
  The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
- `seLinuxOptions` : SELinuxOptions
  SELinuxOptions are the labels to be applied to the container
- `seccompProfile` : SeccompProfile
  SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.
- `windowsOptions` : WindowsSecurityContextOptions
  WindowsSecurityContextOptions contain Windows-specific options and credentials.

AppArmorProfile

AppArmorProfile defines a pod or container's AppArmor settings.

- `localhostProfile` : `string`
  localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
- `type` : `string`
  type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

Capabilities

Adds and removes POSIX capabilities from running containers.

- `add` : `[]string`
  Added capabilities
- `drop` : `[]string`
  Removed capabilities

SELinuxOptions

SELinuxOptions are the labels to be applied to the container

- `level` : `string`
  Level is SELinux level label that applies to the container.
- `role` : `string`
  Role is a SELinux role label that applies to the container.
- `type` : `string`
  Type is a SELinux type label that applies to the container.
- `user` : `string`
  User is a SELinux user label that applies to the container.

SeccompProfile

SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.

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

WindowsSecurityContextOptions

WindowsSecurityContextOptions contain Windows-specific options and credentials.

- `gmsaCredentialSpec` : `string`
  GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ↗) inlines the contents of the GMSA credential spec named by the GMSACredentialSpecName field.
- `gmsaCredentialSpecName` : `string`
  GMSACredentialSpecName is the name of the GMSA credential spec to use.
- `hostProcess` : `boolean`
  HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
- `runAsUserName` : `string`
  The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If

set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

VolumeDevice

volumeDevice describes a mapping of a raw block device within a container.

- `devicePath`: `string`
  devicePath is the path inside of the container that the device will be mapped to.
- `name`: `string`
  name must match the name of a persistentVolumeClaim in the pod

VolumeMount

VolumeMount describes a mounting of a Volume within a container.

- `mountPath`: `string`
  Path within the container at which the volume should be mounted. Must not contain ':'.
- `mountPropagation`: `string`
  mountPropagation determines how mounts are propagated from the host to container and the other way around. When not set, MountPropagationNone is used. This field is beta in 1.10. When RecursiveReadOnly is set to IfPossible or to Enabled, MountPropagation must be None or unspecified (which defaults to None).
- `name`: `string`
  This must match the Name of a Volume.
- `readOnly`: `boolean`
  Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.
- `recursiveReadOnly`: `string`
  RecursiveReadOnly specifies whether read-only mounts should be handled recursively.
  If ReadOnly is false, this field has no meaning and must be unspecified.
  If ReadOnly is true, and this field is set to Disabled, the mount is not made recursively read-only. If this field is set to IfPossible, the mount is made recursively read-only, if it is supported by the container runtime. If this field is set to Enabled, the mount is made

recursively read-only if it is supported by the container runtime, otherwise the pod will not be started and an error will be generated to indicate the reason.
If this field is set to IfPossible or Enabled, MountPropagation must be set to None (or be unspecified, which defaults to None).
If this field is not specified, it is treated as an equivalent of Disabled.

- `subPath` : `string`
  Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).
- `subPathExpr` : `string`
  Expanded path within the volume from which the container's volume should be mounted. Behaves similarly to SubPath but environment variable references $(VAR_NAME) are expanded using the container's environment. Defaults to "" (volume's root). SubPathExpr and SubPath are mutually exclusive.

ObjectReference

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

- `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 ↗

- `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 ↗

Exemplars

- `maxSize` : `integer`

HostAlias

- `hostnames` : `[]string`
- `ip` : `string`

LocalObjectReference

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

- `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 ↗

StatefulSetPersistentVolumeClaimRetentionPolicy

StatefulSetPersistentVolumeClaimRetentionPolicy describes the policy used for PVCs created from the StatefulSet VolumeClaimTemplates.

- `whenDeleted` : `string`
  WhenDeleted specifies what happens to PVCs created from StatefulSet VolumeClaimTemplates when the StatefulSet is deleted. The default policy of `Retain` causes PVCs to not be affected by StatefulSet deletion. The `Delete` policy causes those PVCs to be deleted.
- `whenScaled` : `string`
  WhenScaled specifies what happens to PVCs created from StatefulSet VolumeClaimTemplates when the StatefulSet is scaled down. The default policy of `Retain` causes PVCs to not be affected by a scaledown. The `Delete` policy causes the associated PVCs for any excess pods above the replica count to be deleted.

EmbeddedObjectMetadata

- `annotations` : `map[string]string`
- `labels` : `map[string]string`
- `name` : `string`

PrometheusRuleExcludeConfig

- `ruleName` : `string`
- `ruleNamespace` : `string`

QuerySpec

- `lookbackDelta` : `string`

- `maxConcurrency` : `integer`
- `maxSamples` : `integer`
- `timeout` : `string`

RemoteReadSpec

- `authorization` : Authorization
- `basicAuth` : BasicAuth
- `bearerToken` : `string`
- `bearerTokenFile` : `string`
- `filterExternalLabels` : `boolean`
- `followRedirects` : `boolean`
- `headers` : `map[string]string`
- `name` : `string`
- `noProxy` : `string`
- `oauth2` : OAuth2
- `proxyConnectHeader` : `map[string]array`
- `proxyFromEnvironment` : `boolean`
- `proxyUrl` : `string`
- `readRecent` : `boolean`
- `remoteTimeout` : `string`
- `requiredMatchers` : `map[string]string`
- `tlsConfig` : TLSConfig
- `url` : `string`

OAuth2

- `clientId` : SecretOrConfigMap

- `clientSecret` : SecretKeySelector

  SecretKeySelector selects a key of a Secret.

- `endpointParams` : `map[string]string`
- `noProxy` : `string`
- `proxyConnectHeader` : `map[string]array`
- `proxyFromEnvironment` : `boolean`
- `proxyUrl` : `string`
- `scopes` : `[]string`
- `tlsConfig` : SafeTLSConfig
- `tokenUrl` : `string`

SafeTLSConfig

- `ca` : SecretOrConfigMap
- `cert` : SecretOrConfigMap
- `insecureSkipVerify` : `boolean`
- `keySecret` : SecretKeySelector

  SecretKeySelector selects a key of a Secret.

- `maxVersion` : `string`
- `minVersion` : `string`
- `serverName` : `string`

RemoteWriteSpec

- `authorization` : Authorization
- `azureAd` : AzureAD
- `basicAuth` : BasicAuth
- `bearerToken` : `string`
- `bearerTokenFile` : `string`

- `enableHTTP2` : `boolean`
- `followRedirects` : `boolean`
- `headers` : `map[string]string`
- `metadataConfig` : MetadataConfig
- `name` : `string`
- `noProxy` : `string`
- `oauth2` : OAuth2
- `proxyConnectHeader` : `map[string]array`
- `proxyFromEnvironment` : `boolean`
- `proxyUrl` : `string`
- `queueConfig` : QueueConfig
- `remoteTimeout` : `string`
- `sendExemplars` : `boolean`
- `sendNativeHistograms` : `boolean`
- `sigv4` : Sigv4
- `tlsConfig` : TLSConfig
- `url` : `string`
- `writeRelabelConfigs` : `[]RelabelConfig`

AzureAD

- `cloud` : `string`
- `managedIdentity` : ManagedIdentity
- `oauth` : AzureOAuth
- `sdk` : AzureSDK

ManagedIdentity

- `clientId` : `string`

AzureOAuth

- `clientId` : `string`
- `clientSecret` : SecretKeySelector
  SecretKeySelector selects a key of a Secret.
- `tenantId` : `string`

AzureSDK

- `tenantId` : `string`

MetadataConfig

- `send` : `boolean`
- `sendInterval` : `string`

QueueConfig

- `batchSendDeadline` : `string`
- `capacity` : `integer`
- `maxBackoff` : `string`
- `maxRetries` : `integer`
- `maxSamplesPerSend` : `integer`
- `maxShards` : `integer`
- `minBackoff` : `string`
- `minShards` : `integer`
- `retryOnRateLimit` : `boolean`

- `sampleAgeLimit`: `string`

Rules

- `alert`: RulesAlert

RulesAlert

- `forGracePeriod`: `string`
- `forOutageTolerance`: `string`
- `resendDelay`: `string`

ScrapeClass

- `attachMetadata`: AttachMetadata
- `default`: `boolean`
- `metricRelabelings`: `[]RelabelConfig`
- `name`: `string`
- `relabelings`: `[]RelabelConfig`
- `tlsConfig`: TLSConfig

AttachMetadata

- `node`: `boolean`

PodSecurityContext

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

take precedence over field values of PodSecurityContext.

- `appArmorProfile`: AppArmorProfile
  AppArmorProfile defines a pod or container's AppArmor settings.
- `fsGroup`: `integer`
  A special supplemental group that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod:

  1. The owning GID will be the FSGroup 2. The setgid bit is set (new files created in the volume will be owned by FSGroup) 3. The permission bits are OR'd with rw-rw

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

seLinuxChangePolicy defines how the container's SELinux label is applied to all volumes used by the Pod. It has no effect on nodes that do not support SELinux or to volumes does not support SELinux. Valid values are "MountOption" and "Recursive".
"Recursive" means relabeling of all files on all Pod volumes by the container runtime. This may be slow for large volumes, but allows mixing privileged and unprivileged Pods sharing the same volume on the same node.
"MountOption" mounts all eligible Pod volumes with `-o context` mount option. This requires all Pods that share the same volume to use the same SELinux label. It is not possible to share the same volume among privileged and unprivileged Pods. Eligible volumes are in-tree FibreChannel and iSCSI volumes, and all CSI volumes whose CSI driver announces SELinux support by setting spec.seLinuxMount: true in their CSIDriver instance. Other volumes are always re-labelled recursively. "MountOption" value is allowed only when SELinuxMount feature gate is enabled.
If not specified and SELinuxMount feature gate is enabled, "MountOption" is used. If not specified and SELinuxMount feature gate is disabled, "MountOption" is used for ReadWriteOncePod volumes and "Recursive" for all other volumes.
This field affects only Pods that have SELinux label set, either in PodSecurityContext or in SecurityContext of all containers.
All Pods that use the same volume should use the same seLinuxChangePolicy, otherwise some pods can get stuck in ContainerCreating state. Note that this field cannot be set when spec.os.name is windows.

- `seLinuxOptions`: SELinuxOptions
  SELinuxOptions are the labels to be applied to the container
- `seccompProfile`: SeccompProfile
  SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.
- `supplementalGroups`: `[]integer`
  A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.
- `supplementalGroupsPolicy`: `string`

Defines how supplemental groups of the first container processes are calculated. Valid values are "Merge" and "Strict". If not specified, "Merge" is used. (Alpha) Using the field requires the SupplementalGroupsPolicy feature gate to be enabled and the container runtime must implement support for this feature. Note that this field cannot be set when spec.os.name is windows.

- `sysctls` : `[]Sysctl`
  Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.
- `windowsOptions` : WindowsSecurityContextOptions
  WindowsSecurityContextOptions contain Windows-specific options and credentials.

Sysctl

Sysctl defines a kernel parameter to be set

- `name` : `string`
  Name of a property to set
- `value` : `string`
  Value of a property to set

StorageSpec

- `disableMountSubPath` : `boolean`
- `emptyDir` : EmptyDirVolumeSource
  Represents an empty directory for a pod. Empty directory volumes support ownership management and SELinux relabeling.
- `ephemeral` : EphemeralVolumeSource
  Represents an ephemeral volume that is handled by a normal storage driver.
- `volumeClaimTemplate` : EmbeddedPersistentVolumeClaim
  TypeMeta describes an individual object in an API response or request with strings representing the type of the object and its API schema version. Structures that are versioned or persisted should inline TypeMeta.

EmptyDirVolumeSource

Represents an empty directory for a pod. Empty directory volumes support ownership management and SELinux relabeling.

- `medium` : `string`
  medium represents what type of storage medium should back this directory. The default is "" which means to use the node's default medium. Must be an empty string (default) or Memory. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir ↗
- `sizeLimit` : `string`
  sizeLimit is the total amount of local storage required for this EmptyDir volume. The size limit is also applicable for memory medium. The maximum usage on memory medium EmptyDir would be the minimum value between the SizeLimit specified here and the sum of memory limits of all containers in a pod. The default is nil which means that the limit is undefined. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir ↗

EphemeralVolumeSource

Represents an ephemeral volume that is handled by a normal storage driver.

- `volumeClaimTemplate` : PersistentVolumeClaimTemplate
  PersistentVolumeClaimTemplate is used to produce PersistentVolumeClaim objects as part of an EphemeralVolumeSource.

PersistentVolumeClaimTemplate

PersistentVolumeClaimTemplate is used to produce PersistentVolumeClaim objects as part of an EphemeralVolumeSource.

- `metadata` : ObjectMeta ↗
  ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
- `spec` : PersistentVolumeClaimSpec

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

PersistentVolumeClaimSpec

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

- `accessModes` : `[]string`
  accessModes contains the desired access modes the volume should have. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1 ↗
- `dataSource` : TypedLocalObjectReference
  TypedLocalObjectReference contains enough information to let you locate the typed referenced object inside the same namespace.
- `dataSourceRef` : TypedObjectReference
  TypedObjectReference contains enough information to let you locate the typed referenced object
- `resources` : VolumeResourceRequirements
  VolumeResourceRequirements describes the storage resource requirements for a volume.
- `selector` : LabelSelector
  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.
- `storageClassName` : `string`
  storageClassName is the name of the StorageClass required by the claim. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#class-1 ↗
- `volumeAttributesClassName` : `string`
  volumeAttributesClassName may be used to set the VolumeAttributesClass used by this claim. If specified, the CSI driver will create or update the volume with the attributes defined in the corresponding VolumeAttributesClass. This has a different purpose than storageClassName, it can be changed after the claim is created. An empty string value means that no VolumeAttributesClass will be applied to the claim but it's not allowed to reset this field to empty string once it is set. If unspecified and the PersistentVolumeClaim is unbound, the default VolumeAttributesClass will be set by the persistentvolume controller

if it exists. If the resource referred to by volumeAttributesClass does not exist, this PersistentVolumeClaim will be set to a Pending state, as reflected by the modifyVolumeStatus field, until such as a resource exists. More info: https://kubernetes.io/docs/concepts/storage/volume-attributes-classes/ ↗ (Beta) Using this field requires the VolumeAttributesClass feature gate to be enabled (off by default).

- `volumeMode` : `string`
  volumeMode defines what type of volume is required by the claim. Value of Filesystem is implied when not included in claim spec.
- `volumeName` : `string`
  volumeName is the binding reference to the PersistentVolume backing this claim.

TypedLocalObjectReference

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

- `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

TypedObjectReference

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

- `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.

VolumeResourceRequirements

VolumeResourceRequirements describes the storage resource requirements for a volume.

- `limits` : `map[string]Quantity`
  Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗
- `requests` : `map[string]Quantity`
  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/ ↗

EmbeddedPersistentVolumeClaim

TypeMeta describes an individual object in an API response or request with strings representing the type of the object and its API schema version. Structures that are versioned or persisted should inline TypeMeta.

- `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` : EmbeddedObjectMetadata
- `spec` : PersistentVolumeClaimSpec
  PersistentVolumeClaimSpec describes the common attributes of storage devices and allows a Source for provider-specific attributes
- `status` : PersistentVolumeClaimStatus
  PersistentVolumeClaimStatus is the current status of a persistent volume claim.

PersistentVolumeClaimStatus

PersistentVolumeClaimStatus is the current status of a persistent volume claim.

- `accessModes` : `[]string`
  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 ↗
- `allocatedResourceStatuses` : `map[string]string`
  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.

- `allocatedResources` : `map[string]Quantity`
  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.
- `capacity` : `map[string]Quantity`
  capacity represents the actual resources of the underlying volume.
- `conditions` : `[]PersistentVolumeClaimCondition`

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

- `currentVolumeAttributesClassName` : `string`
  currentVolumeAttributesClassName is the current name of the VolumeAttributesClass the PVC is using. When unset, there is no VolumeAttributeClass applied to this PersistentVolumeClaim This is a beta field and requires enabling VolumeAttributesClass feature (off by default).
- `modifyVolumeStatus` : ModifyVolumeStatus
  ModifyVolumeStatus represents the status object of ControllerModifyVolume operation
- `phase` : `string`
  phase represents the current phase of PersistentVolumeClaim.

PersistentVolumeClaimCondition

PersistentVolumeClaimCondition contains details about state of pvc

- `lastProbeTime` : `string`
  lastProbeTime is the time we probed the condition.
- `lastTransitionTime` : `string`
  lastTransitionTime is the time the condition transitioned from one status to another.
- `message` : `string`
  message is the human-readable message indicating details about last transition.
- `reason` : `string`
  reason is a unique, this should be a short, machine understandable string that gives the reason for condition's last transition. If it reports "Resizing" that means the underlying persistent volume is being resized.
- `status` : `string`
  Status is the status of the condition. Can be True, False, Unknown. More info: https://kubernetes.io/docs/reference/kubernetes-api/config-and-storage-resources/persistent-volume-claim-v1/#:~:text=state%20of%20pvc-,conditions.status,-(string)%2C%20required ↗
- `type` : `string`

Type is the type of the condition. More info: https://kubernetes.io/docs/reference/kubernetes-api/config-and-storage-resources/persistent-volume-claim-v1/#:~:text=set%20to%20%27ResizeStarted%27.-,PersistentVolumeClaimCondition,-contains%20details%20about ↗

ModifyVolumeStatus

ModifyVolumeStatus represents the status object of ControllerModifyVolume operation

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

ThanosSpec

- `additionalArgs` : `[]Argument`
- `baseImage` : `string`
- `blockSize` : `string`
- `getConfigInterval` : `string`
- `getConfigTimeout` : `string`

- `grpcListenLocal` : `boolean`
- `grpcServerTlsConfig` : TLSConfig
- `httpListenLocal` : `boolean`
- `image` : `string`
- `listenLocal` : `boolean`
- `logFormat` : `string`
- `logLevel` : `string`
- `minTime` : `string`
- `objectStorageConfig` : SecretKeySelector
  SecretKeySelector selects a key of a Secret.
- `objectStorageConfigFile` : `string`
- `readyTimeout` : `string`
- `resources` : ResourceRequirements
  ResourceRequirements describes the compute resource requirements.
- `sha` : `string`
- `tag` : `string`
- `tracingConfig` : SecretKeySelector
  SecretKeySelector selects a key of a Secret.
- `tracingConfigFile` : `string`
- `version` : `string`
- `volumeMounts` : `[]VolumeMount`

Toleration

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

- `effect` : `string`
  Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.

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

TopologySpreadConstraint

- `additionalLabelSelectors` : `string`
- `labelSelector` : LabelSelector
  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.
- `matchLabelKeys` : `[]string`
- `maxSkew` : `integer`
- `minDomains` : `integer`
- `nodeAffinityPolicy` : `string`
- `nodeTaintsPolicy` : `string`
- `topologyKey` : `string`
- `whenUnsatisfiable` : `string`

PrometheusTracingConfig

- `clientType` : `string`
- `compression` : `string`
- `endpoint` : `string`
- `headers` : `map[string]string`
- `insecure` : `boolean`
- `samplingFraction` : `string`
- `timeout` : `string`
- `tlsConfig` : TLSConfig

TSDBSpec

- `outOfOrderTimeWindow` : `string`

Volume

Volume represents a named volume in a pod that may be accessed by any container in the pod.

- `awsElasticBlockStore` : AWSElasticBlockStoreVolumeSource
  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.
- `azureDisk` : AzureDiskVolumeSource
  AzureDisk represents an Azure Data Disk mount on the host and bind mount to the pod.
- `azureFile` : AzureFileVolumeSource
  AzureFile represents an Azure File Service mount on the host and bind mount to the pod.
- `cephfs` : CephFSVolumeSource

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

- `cinder` : CinderVolumeSource
  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.
- `configMap` : ConfigMapVolumeSource
  Adapts a ConfigMap into a volume.
  The contents of the target ConfigMap's Data field will be presented in a volume as files using the keys in the Data field as the file names, unless the items element is populated with specific mappings of keys to paths. ConfigMap volumes support ownership management and SELinux relabeling.
- `csi` : CSIVolumeSource
  Represents a source location of a volume to mount, managed by an external CSI driver
- `downwardAPI` : DownwardAPIVolumeSource
  DownwardAPIVolumeSource represents a volume containing downward API info. Downward API volumes support ownership management and SELinux relabeling.
- `emptyDir` : EmptyDirVolumeSource
  Represents an empty directory for a pod. Empty directory volumes support ownership management and SELinux relabeling.
- `ephemeral` : EphemeralVolumeSource
  Represents an ephemeral volume that is handled by a normal storage driver.
- `fc` : FCVolumeSource
  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.
- `flexVolume` : FlexVolumeSource
  FlexVolume represents a generic volume resource that is provisioned/attached using an exec based plugin.
- `flocker` : FlockerVolumeSource
  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.
- `gcePersistentDisk` : GCEPersistentDiskVolumeSource

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.

- `gitRepo` : GitRepoVolumeSource
  Represents a volume that is populated with the contents of a git repository. Git repo volumes do not support ownership management. Git repo volumes support SELinux relabeling.
  DEPRECATED: GitRepo is deprecated. To provision a container with a git repo, mount an EmptyDir into an InitContainer that clones the repo using git, then mount the EmptyDir into the Pod's container.
- `glusterfs` : GlusterfsVolumeSource
  Represents a Glusterfs mount that lasts the lifetime of a pod. Glusterfs volumes do not support ownership management or SELinux relabeling.
- `hostPath` : HostPathVolumeSource
  Represents a host path mapped into a pod. Host path volumes do not support ownership management or SELinux relabeling.
- `image` : ImageVolumeSource
  ImageVolumeSource represents a image volume resource.
- `iscsi` : ISCSIVolumeSource
  Represents an ISCSI disk. ISCSI volumes can only be mounted as read/write once. ISCSI volumes support ownership management and SELinux relabeling.
- `name` : `string`
  name of the volume. Must be a DNS_LABEL and unique within the pod. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
- `nfs` : NFSVolumeSource
  Represents an NFS mount that lasts the lifetime of a pod. NFS volumes do not support ownership management or SELinux relabeling.
- `persistentVolumeClaim` : PersistentVolumeClaimVolumeSource
  PersistentVolumeClaimVolumeSource references the user's PVC in the same namespace. This volume finds the bound PV and mounts that volume for the pod. A PersistentVolumeClaimVolumeSource is, essentially, a wrapper around another type of volume that is owned by someone else (the system).

- `photonPersistentDisk` : PhotonPersistentDiskVolumeSource
  Represents a Photon Controller persistent disk resource.
- `portworxVolume` : PortworxVolumeSource
  PortworxVolumeSource represents a Portworx volume resource.
- `projected` : ProjectedVolumeSource
  Represents a projected volume source
- `quobyte` : QuobyteVolumeSource
  Represents a Quobyte mount that lasts the lifetime of a pod. Quobyte volumes do not support ownership management or SELinux relabeling.
- `rbd` : RBDVolumeSource
  Represents a Rados Block Device mount that lasts the lifetime of a pod. RBD volumes support ownership management and SELinux relabeling.
- `scaleIO` : ScaleIOVolumeSource
  ScaleIOVolumeSource represents a persistent ScaleIO volume
- `secret` : SecretVolumeSource
  Adapts a Secret into a volume.
  The contents of the target Secret's Data field will be presented in a volume as files using the keys in the Data field as the file names. Secret volumes support ownership management and SELinux relabeling.
- `storageos` : StorageOSVolumeSource
  Represents a StorageOS persistent volume resource.
- `vsphereVolume` : VsphereVirtualDiskVolumeSource
  Represents a vSphere volume resource.

AWSElasticBlockStoreVolumeSource

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.

- `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#awselasticblockstore ↗

- `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 volume partition for /dev/sda is "0" (or you can leave the property empty).
- `readOnly` : `boolean`
  readOnly value true will force the readOnly setting in VolumeMounts. More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore ↗
- `volumeID` : `string`
  volumeID is unique ID of the persistent disk resource in AWS (Amazon EBS volume). More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore ↗

AzureDiskVolumeSource

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

- `cachingMode` : `string`
  cachingMode is the Host Caching mode: None, Read Only, Read Write.
- `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
- `readOnly` : `boolean`

readOnly Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

AzureFileVolumeSource

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

- `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
- `shareName` : `string`
  shareName is the azure share Name

CephFSVolumeSource

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

- `monitors` : `[]string`
  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: 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`: LocalObjectReference
  LocalObjectReference contains enough information to let you locate the referenced object inside the same 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 ↗

CinderVolumeSource

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.

- `fsType`: `string`
  fsType is the 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 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`: LocalObjectReference
  LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
- `volumeID`: `string`
  volumeID used to identify the volume in cinder. More info: https://examples.k8s.io/mysql-cinder-pd/README.md ↗

ConfigMapVolumeSource

Adapts a ConfigMap into a volume.

The contents of the target ConfigMap's Data field will be presented in a volume as files using the keys in the Data field as the file names, unless the items element is populated with

specific mappings of keys to paths. ConfigMap volumes support ownership management and SELinux relabeling.

- `defaultMode` : `integer`
  defaultMode is optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
- `items` : `[]KeyToPath`
  items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
- `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`
  optional specify whether the ConfigMap or its keys must be defined

KeyToPath

Maps a string key to a path within a volume.

- `key` : `string`
  key is the key to project.
- `mode` : `integer`
  mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the

volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

- `path` : `string`
  path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

CSIVolumeSource

Represents a source location of a volume to mount, managed by an external CSI driver

- `driver` : `string`
  driver is the name of the CSI driver that handles this volume. Consult with your admin for the correct name as registered in the cluster.
- `fsType` : `string`
  fsType to mount. Ex. "ext4", "xfs", "ntfs". If not provided, the empty value is passed to the associated CSI driver which will determine the default filesystem to apply.
- `nodePublishSecretRef` : LocalObjectReference
  LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
- `readOnly` : `boolean`
  readOnly specifies a read-only configuration for the volume. Defaults to false (read/write).
- `volumeAttributes` : `map[string]string`
  volumeAttributes stores driver-specific properties that are passed to the CSI driver. Consult your driver's documentation for supported values.

DownwardAPIVolumeSource

DownwardAPIVolumeSource represents a volume containing downward API info. Downward API volumes support ownership management and SELinux relabeling.

- `defaultMode` : `integer`
  Optional: mode bits to use on created files by default. Must be a Optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and

0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

- `items` : `[]DownwardAPIVolumeFile`
  Items is a list of downward API volume file

DownwardAPIVolumeFile

DownwardAPIVolumeFile represents information to create the file containing the pod field

- `fieldRef` : ObjectFieldSelector
  ObjectFieldSelector selects an APIVersioned field of an object.
- `mode` : `integer`
  Optional: mode bits used to set permissions on this file, must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
- `path` : `string`
  Required: Path is the relative path name of the file to be created. Must not be absolute or contain the '..' path. Must be utf-8 encoded. The first item of the relative path must not start with '..'
- `resourceFieldRef` : ResourceFieldSelector
  ResourceFieldSelector represents container resources (cpu, memory) and their output format

FCVolumeSource

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.

- `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` : `[]string`
  targetWWNs is Optional: FC target worldwide names (WWNs)
- `wwids` : `[]string`
  wwids Optional: FC volume world wide identifiers (wwids) Either wwids or combination of targetWWNs and lun must be set, but not both simultaneously.

FlexVolumeSource

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

- `driver` : `string`
  driver is the name of the driver to use for this volume.
- `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` : `map[string]string`
  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` : LocalObjectReference
  LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

FlockerVolumeSource

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.

- `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

GCEPersistentDiskVolumeSource

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.

- `fsType` : `string`
  fsType is 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#gcepersistentdisk ↗
- `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 volume partition for /dev/sda is "0" (or you can leave the property empty). More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk ↗
- `pdName` : `string`
  pdName is unique name of the PD resource in GCE. Used to identify the disk in GCE. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk ↗
- `readOnly` : `boolean`

readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk ↗

GitRepoVolumeSource

Represents a volume that is populated with the contents of a git repository. Git repo volumes do not support ownership management. Git repo volumes support SELinux relabeling.

DEPRECATED: GitRepo is deprecated. To provision a container with a git repo, mount an EmptyDir into an InitContainer that clones the repo using git, then mount the EmptyDir into the Pod's container.

- `directory`: `string`
  directory is the target directory name. Must not contain or start with '..'. If '.' is supplied, the volume directory will be the git repository. Otherwise, if specified, the volume will contain the git repository in the subdirectory with the given name.
- `repository`: `string`
  repository is the URL
- `revision`: `string`
  revision is the commit hash for the specified revision.

GlusterfsVolumeSource

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

- `endpoints`: `string`
  endpoints is the endpoint name that details Glusterfs topology. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod ↗
- `path`: `string`
  path is the Glusterfs volume path. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod ↗
- `readOnly`: `boolean`

readOnly here will force the Glusterfs volume to be mounted with read-only permissions. Defaults to false. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod ↗

HostPathVolumeSource

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

- `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 ↗

ImageVolumeSource

ImageVolumeSource represents a image volume resource.

- `pullPolicy`: `string`
  Policy for pulling OCI objects. Possible values are: Always: the kubelet always attempts to pull the reference. Container creation will fail If the pull fails. Never: the kubelet never pulls the reference and only uses a local image or artifact. Container creation will fail if the reference isn't present. IfNotPresent: the kubelet pulls if the reference isn't already present on disk. Container creation will fail if the reference isn't present and the pull fails. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise.
- `reference`: `string`
  Required: Image or artifact reference to be used. Behaves in the same way as pod.spec.containers[*].image. Pull secrets will be assembled in the same way as for the container image by looking up node credentials, SA image pull secrets, and pod spec image pull secrets. More info: https://kubernetes.io/docs/concepts/containers/images ↗
  This field is optional to allow higher level config management to default or override container images in workload controllers like Deployments and StatefulSets.

ISCSIVolumeSource

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

- `chapAuthDiscovery` : `boolean`
  chapAuthDiscovery defines whether support iSCSI Discovery CHAP authentication
- `chapAuthSession` : `boolean`
  chapAuthSession defines whether support iSCSI Session CHAP authentication
- `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#iscsi ↗
- `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.
- `iqn` : `string`
  iqn is the target iSCSI Qualified Name.
- `iscsiInterface` : `string`
  iscsiInterface is the interface Name that uses an iSCSI transport. Defaults to 'default' (tcp).
- `lun` : `integer`
  lun represents iSCSI Target Lun number.
- `portals` : `[]string`
  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` : LocalObjectReference
  LocalObjectReference contains enough information to let you locate the referenced object inside the same 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).

NFSVolumeSource

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

- `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 ↗

PersistentVolumeClaimVolumeSource

PersistentVolumeClaimVolumeSource references the user's PVC in the same namespace. This volume finds the bound PV and mounts that volume for the pod. A PersistentVolumeClaimVolumeSource is, essentially, a wrapper around another type of volume that is owned by someone else (the system).

- `claimName` : `string`
  claimName is the name of a PersistentVolumeClaim in the same namespace as the pod using this volume. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims ↗
- `readOnly` : `boolean`
  readOnly Will force the ReadOnly setting in VolumeMounts. Default false.

PhotonPersistentDiskVolumeSource

Represents a Photon Controller persistent disk resource.

- `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

PortworxVolumeSource

PortworxVolumeSource represents a Portworx volume resource.

- `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.
- `volumeID` : `string`
  volumeID uniquely identifies a Portworx volume

ProjectedVolumeSource

Represents a projected volume source

- `defaultMode` : `integer`
  defaultMode are the mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
- `sources` : `[]VolumeProjection`
  sources is the list of volume projections. Each entry in this list handles one source.

VolumeProjection

Projection that may be projected along with other supported volume types. Exactly one of these fields must be set.

- `clusterTrustBundle`: ClusterTrustBundleProjection
  ClusterTrustBundleProjection describes how to select a set of ClusterTrustBundle objects and project their contents into the pod filesystem.
- `configMap`: ConfigMapProjection
  Adapts a ConfigMap into a projected volume.
  The contents of the target ConfigMap's Data field will be presented in a projected volume as files using the keys in the Data field as the file names, unless the items element is populated with specific mappings of keys to paths. Note that this is identical to a configmap volume source without the default mode.
- `downwardAPI`: DownwardAPIProjection
  Represents downward API info for projecting into a projected volume. Note that this is identical to a downwardAPI volume source without the default mode.
- `secret`: SecretProjection
  Adapts a secret into a projected volume.
  The contents of the target Secret's Data field will be presented in a projected volume as files using the keys in the Data field as the file names. Note that this is identical to a secret volume source without the default mode.
- `serviceAccountToken`: ServiceAccountTokenProjection
  ServiceAccountTokenProjection represents a projected service account token volume. This projection can be used to insert a service account token into the pods runtime filesystem for use against APIs (Kubernetes API Server or otherwise).

ClusterTrustBundleProjection

ClusterTrustBundleProjection describes how to select a set of ClusterTrustBundle objects and project their contents into the pod filesystem.

- `labelSelector`: LabelSelector

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.

- `name` : `string`
  Select a single ClusterTrustBundle by object name. Mutually-exclusive with signerName and labelSelector.
- `optional` : `boolean`
  If true, don't block pod startup if the referenced ClusterTrustBundle(s) aren't available. If using name, then the named ClusterTrustBundle is allowed not to exist. If using signerName, then the combination of signerName and labelSelector is allowed to match zero ClusterTrustBundles.
- `path` : `string`
  Relative path from the volume root to write the bundle.
- `signerName` : `string`
  Select all ClusterTrustBundles that match this signer name. Mutually-exclusive with name. The contents of all selected ClusterTrustBundles will be unified and deduplicated.

ConfigMapProjection

Adapts a ConfigMap into a projected volume.

The contents of the target ConfigMap's Data field will be presented in a projected volume as files using the keys in the Data field as the file names, unless the items element is populated with specific mappings of keys to paths. Note that this is identical to a configmap volume source without the default mode.

- `items` : `[]KeyToPath`
  items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
- `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`
  optional specify whether the ConfigMap or its keys must be defined

DownwardAPIProjection

Represents downward API info for projecting into a projected volume. Note that this is identical to a downwardAPI volume source without the default mode.

- `items` : `[]DownwardAPIVolumeFile`
  Items is a list of DownwardAPIVolume file

SecretProjection

Adapts a secret into a projected volume.

The contents of the target Secret's Data field will be presented in a projected volume as files using the keys in the Data field as the file names. Note that this is identical to a secret volume source without the default mode.

- `items` : `[]KeyToPath`
  items if unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
- `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`
  optional field specify whether the Secret or its key must be defined

ServiceAccountTokenProjection

ServiceAccountTokenProjection represents a projected service account token volume. This projection can be used to insert a service account token into the pods runtime filesystem for use against APIs (Kubernetes API Server or otherwise).

- `audience` : `string`
  audience is the intended audience of the token. A recipient of a token must identify itself with an identifier specified in the audience of the token, and otherwise should reject the token. The audience defaults to the identifier of the apiserver.
- `expirationSeconds` : `integer`
  expirationSeconds is the requested duration of validity of the service account token. As the token approaches expiration, the kubelet volume plugin will proactively rotate the service account token. The kubelet will start trying to rotate the token if the token is older than 80 percent of its time to live or if the token is older than 24 hours.Defaults to 1 hour and must be at least 10 minutes.
- `path` : `string`
  path is the path relative to the mount point of the file to project the token into.

QuobyteVolumeSource

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

- `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, 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.

RBDVolumeSource

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

- `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 ↗
- `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` : `[]string`
  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` : LocalObjectReference
  LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
- `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 ↗

ScaleIOVolumeSource

ScaleIOVolumeSource represents a persistent ScaleIO volume

- `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.
- `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` : LocalObjectReference
  LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
- `sslEnabled` : `boolean`
  sslEnabled Flag 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.
- `volumeName` : `string`
  volumeName is the name of a volume already created in the ScaleIO system that is associated with this volume source.

SecretVolumeSource

Adapts a Secret into a volume.

The contents of the target Secret's Data field will be presented in a volume as files using the keys in the Data field as the file names. Secret volumes support ownership management and SELinux relabeling.

- `defaultMode` : `integer`
  defaultMode is Optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
- `items` : `[]KeyToPath`
  items If unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
- `optional` : `boolean`
  optional field specify whether the Secret or its keys must be defined
- `secretName` : `string`
  secretName is the name of the secret in the pod's namespace to use. More info: https://kubernetes.io/docs/concepts/storage/volumes#secret ↗

StorageOSVolumeSource

Represents a StorageOS persistent volume resource.

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

VsphereVirtualDiskVolumeSource

Represents a vSphere volume resource.

- `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.
- `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

PrometheusWebSpec

- `httpConfig` : WebHTTPConfig
- `maxConnections` : `integer`
- `pageTitle` : `string`
- `tlsConfig` : WebTLSConfig

WebHTTPConfig

- `headers` : WebHTTPHeaders
- `http2` : `boolean`

WebHTTPHeaders

- `contentSecurityPolicy` : `string`
- `strictTransportSecurity` : `string`
- `xContentTypeOptions` : `string`
- `xFrameOptions` : `string`
- `xXSSProtection` : `string`

WebTLSConfig

- `cert` : SecretOrConfigMap
- `cipherSuites` : `[]string`
- `clientAuthType` : `string`

- `client_ca` : SecretOrConfigMap
- `curvePreferences` : `[]string`
- `keySecret` : SecretKeySelector

  SecretKeySelector selects a key of a Secret.
- `maxVersion` : `string`
- `minVersion` : `string`
- `preferServerCipherSuites` : `boolean`

PrometheusStatus

- `availableReplicas` : `integer`
- `conditions` : `[]Condition`
- `paused` : `boolean`
- `replicas` : `integer`
- `selector` : `string`
- `shardStatuses` : `[]ShardStatus`
- `shards` : `integer`
- `unavailableReplicas` : `integer`
- `updatedReplicas` : `integer`

Condition

- `lastTransitionTime` : `string`
- `message` : `string`
- `observedGeneration` : `integer`
- `reason` : `string`
- `status` : `string`
- `type` : `string`

ShardStatus

- `availableReplicas` : `integer`
- `replicas` : `integer`
- `shardID` : `string`
- `unavailableReplicas` : `integer`
- `updatedReplicas` : `integer`

PrometheusRule

- `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.
- `spec` : PrometheusRuleSpec

PrometheusRuleSpec

- `groups` : `[]RuleGroup`

RuleGroup

- `interval` : `string`
- `limit` : `integer`
- `name` : `string`
- `partial_response_strategy` : `string`
- `rules` : `[]Rule`

Rule

- `alert` : `string`
- `annotations` : `map[string]string`
- `expr` : `string`
- `for` : `string`
- `keep_firing_for` : `string`
- `labels` : `map[string]string`
- `record` : `string`

Topology

- `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.

- `rootNode` : TopologyRootNode
  ObjectReference contains enough information to let you inspect or modify the referred object.

TopologyRootNode

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

- `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 ↗
- `namespace` : `string`
  Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
- `parent` : ObjectReference
  ObjectReference contains enough information to let you inspect or modify the referred object.
- `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 ↗

ApplicationSetTopology

- `ResourceNodes` : `map[string]object`
- `RootNode` : Node
- `rootNodeType` : `string`

NodeMap

Node

- `apiVersion` : `string`
- `cluster` : `string`
- `component` : `string`
- `displayName` : `string`
- `edges` : `[]Edge`
- `kind` : `string`
- `name` : `string`
- `namespace` : `string`
- `originalResource` : OriginalResource
- `restricted` : `boolean`
- `state` : `string`

- `syncStatus` : `string`
- `type` : `string`
- `uid` : `string`

Edge

- `from` : `string`
- `to` : `string`
- `type` : `string`

OriginalResource