

Kafka APIs

[RdsKafka](#)

[RdsTopic](#)

[RdsKafkaU](#)

[RdsMirrorMaker2](#)

RdsKafka

Description

RdsKafka is the Schema for the rdskafkas API

Type

object

Specification

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

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

.spec

Description

RdsKafkaSpec defines the desired state of RdsKafka

Type

object

Property	Type	Description
config	object	Custom configuration properties

Property	Type	Description
<code>controller</code>	<code>object</code>	Controller configuration for KRaft mode
<code>entityOperator</code>	<code>object</code>	EntityOperator configuration
<code>ipFamilyPrefer</code>	<code>string</code>	Network family preference for brokers
<code>kafka</code>	<code>object</code>	Kafka configuration
<code>kafkaExporter</code>	<code>object</code>	Kafka exporter configuration
<code>mode</code>	<code>string</code>	Mode field defines the mode for kafka cluster
<code>pause</code>	<code>boolean</code>	Pause field indicates whether Kafka is paused.
<code>replicas</code>	<code>integer</code>	Number of replicas for Kafka brokers
<code>resources</code>	<code>object</code>	Resource requests/limits for pods
<code>storage</code>	<code>object</code>	Storage configuration for persistent volumes

Property	Type	Description
<code>upgradeOption</code>	<code>object</code>	UpgradeOption defines the upgrade strategy for the Kafka instance.
<code>version</code>	<code>string</code>	Kafka Cluster version
<code>zookeeper</code>	<code>object</code>	Zookeeper configuration

.spec.config

Description

Custom configuration properties

Type

`object`

.spec.controller

Description

Controller configuration for KRaft mode

Type

`object`

Property	Type	Description
<code>jvmOptions</code>	<code>object</code>	JVM options such as Xms and Xmx for Kafka brokers.

Property	Type	Description
<code>replicas</code>	<code>integer</code>	Replicas of KRaft controllers
<code>resources</code>	<code>object</code>	Resource requirements for controller pods
<code>roles</code>	<code>array</code>	Roles assigned to controller nodes
<code>storage</code>	<code>object</code>	Storage configuration for controller data
<code>template</code>	<code>object</code>	Pod template for brokers

`.spec.controller.jvmOptions`

Description

JVM options such as Xms and Xmx for Kafka brokers.

Type

`object`

Property	Type	Description
<code>-Xms</code>	<code>string</code>	The Xms field sets the JVM min heap size parameter
<code>-Xmx</code>	<code>string</code>	The Xmx field sets the JVM max heap size parameter

.spec.controller.resources

Description

Resource requirements for controller pods

Type

object

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

.spec.controller.resources.claims

Description

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

Type

array

`.spec.controller.resources.claims[]`

Description

ResourceClaim references one entry in `PodSpec.ResourceClaims`.

Type

object

Required

name

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

`.spec.controller.resources.limits`

Description

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

Type

object

.spec.controller.resources.requests**Description**

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

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

Type

object

.spec.controller.roles**Description**

Roles assigned to controller nodes

Type

array

.spec.controller.roles[]**Type**

string

.spec.controller.storage**Description**

Storage configuration for controller data

Type

object

Required

class

size

Property	Type	Description
class	string	
deleteClaim	boolean	
size		The name of the StorageClass to claim a PersistentVolume from.

.spec.controller.template

Description

Pod template for brokers

Type

object

Property	Type	Description
pod	object	Pod template for containers in the pod. Contains security context, affinity and tolerations settings.

.spec.controller.template.pod

Description

Pod template for containers in the pod. Contains security context, affinity and tolerations settings.

Type

object

Property	Type	Description
<code>affinity</code>	<code>object</code>	Affinity and anti-affinity rules for pod scheduling.
<code>enableServiceLinks</code>	<code>boolean</code>	EnableServiceLinks Indicates whether information about services should be injected into Pod's environment variables.
<code>securityContext</code>	<code>object</code>	Security context for pods such as permissions and privilege escalation.
<code>tolerations</code>	<code>array</code>	Tolerations for pods to schedule onto nodes with taints.

`.spec.controller.template.pod.affinity`

Description

Affinity and anti-affinity rules for pod scheduling.

Type

`object`

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

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

`.spec.controller.template.pod.affinity.nodeAffinity`

Description

Describes node affinity scheduling rules for the pod.

Type

`object`

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

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

`.spec.controller.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

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

Type

array

`.spec.controller.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

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

Type

object

Required

preference

weight

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

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

`.spec.controller.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference`

Description

A node selector term, associated with the corresponding weight.

Type

`object`

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

`.spec.controller.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions`

Description

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

Type

`array`

`.spec.controller.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[]`

Description

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

Type

object

Required

key

operator

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

`.spec.controller.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.`

matchExpressions[].values

Description

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

Type

array

.spec.controller.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values[]

Type

string

.spec.controller.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields

Description

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

Type

array

.spec.controller.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[]

Description

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

Type

object

Required

key

operator

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

`.spec.controller.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values`

Description

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

Type

array

.spec.controller.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values[]

Type

string

.spec.controller.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

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

Type

object

Required

nodeSelectorTerms

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

.spec.controller.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorT

erms

Description

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

Type

array

`.spec.controller.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms`

Description

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

Type

object

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

`.spec.controller.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions`

Description

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

Type

array

`.spec.controller.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[]`

Description

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

Type

object

Required

key

operator

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

.spec.controller.template.pod.affinity.nodeAffinity.required DuringSchedulingIgnoredDuringExecution.nodeSelectorT erms[].matchExpressions[].values

Description

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

Type

array

.spec.controller.template.pod.affinity.nodeAffinity.required DuringSchedulingIgnoredDuringExecution.nodeSelectorT erms[].matchExpressions[].values[]

Type

string

.spec.controller.template.pod.affinity.nodeAffinity.required DuringSchedulingIgnoredDuringExecution.nodeSelectorT erms[].matchFields

Description

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

Type

array

.spec.controller.template.pod.affinity.nodeAffinity.required DuringSchedulingIgnoredDuringExecution.nodeSelectorT

terms[].matchFields[]

Description

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

Type

object

Required

key

operator

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

.spec.controller.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values

Description

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

Type

array

.spec.controller.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values[]

Type

string

.spec.controller.template.pod.affinity.podAffinity

Description

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

Type

object

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, it may choose a node that violates one or more the expressions. The node that is most

Property	Type	Description
		<p>preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution affinity expressions, etc.) we compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has a podAffinityTerm which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.</p>
<p><code>requiredDuringSchedulingIgnoredDuringExecution</code></p>	<p><code>array</code></p>	<p>If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to pod label update), the system may or may not try to eventually evict the pod from its node. When</p>

Property	Type	Description
		there are multiple elements, the lists of nodes corresponding each podAffinityTerm intersected, i.e. all terms must be satisfied.

.spec.controller.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

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

Type

array

.spec.controller.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

Description

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

Type

object

Required

podAffinityTerm

weight

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

.spec.controller.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm

Description

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

Type

object

Required

topologyKey

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

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

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

`.spec.controller.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector`

Description

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

Type

object

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

.spec.controller.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions

Description

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

Type

array

.spec.controller.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]

Description

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

Type

object

Required

key

operator

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

`.spec.controller.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values`

Description

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

Type

array

`.spec.controller.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]`

Type

string

`.spec.controller.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels`

Description

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

Type

object

`.spec.controller.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys`

Description

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

Type

`array`

**.spec.controller.template.pod.affinity.podAffinity.preferred
DuringSchedulingIgnoredDuringExecution[].podAffinityTe
rm.matchLabelKeys[]**

Type

`string`

**.spec.controller.template.pod.affinity.podAffinity.preferred
DuringSchedulingIgnoredDuringExecution[].podAffinityTe
rm.mismatchLabelKeys**

Description

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

Type

`array`

**.spec.controller.template.pod.affinity.podAffinity.preferred
DuringSchedulingIgnoredDuringExecution[].podAffinityTe
rm.mismatchLabelKeys[]**

Type

`string`

`.spec.controller.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector`

Description

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

Type

object

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

`.spec.controller.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions`

Description

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

Type

array

`.spec.controller.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]`

Description

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

Type

object

Required

key

operator

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

`.spec.controller.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTe`

`rm.namespaceSelector.matchExpressions[].values`

Description

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

Type

array

`.spec.controller.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]`

Type

string

`.spec.controller.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels`

Description

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

Type

object

`.spec.controller.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces`

Description

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

Type

array

.spec.controller.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

string

.spec.controller.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

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

Type

array

.spec.controller.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

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

Type

object

Required

`topologyKey`

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

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

Property	Type	Description
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.controller.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector`

Description

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

Type

`object`

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

`.spec.controller.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

Description

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

Type

array

`.spec.controller.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]`

Description

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

Type

object

Required

key

operator

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

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

.spec.controller.template.pod.affinity.podAffinity.required DuringSchedulingIgnoredDuringExecution[].labelSelector. matchExpressions[].values

Description

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

Type

array

.spec.controller.template.pod.affinity.podAffinity.required DuringSchedulingIgnoredDuringExecution[].labelSelector. matchExpressions[].values[]

Type

string

.spec.controller.template.pod.affinity.podAffinity.required DuringSchedulingIgnoredDuringExecution[].labelSelector. matchLabels

Description

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

Type

object

.spec.controller.template.pod.affinity.podAffinity.required DuringSchedulingIgnoredDuringExecution[].matchLabelK eys

Description

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

Type

array

.spec.controller.template.pod.affinity.podAffinity.required DuringSchedulingIgnoredDuringExecution[].matchLabelK eys[]

Type

string

.spec.controller.template.pod.affinity.podAffinity.required DuringSchedulingIgnoredDuringExecution[].mismatchLab

elKeys

Description

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

Type

array

`.spec.controller.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]`

Type

string

`.spec.controller.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector`

Description

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

Type

object

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

`.spec.controller.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions`

Description

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

Type

array

`.spec.controller.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]`

Description

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

Type

object

Required

key

operator

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

.spec.controller.template.pod.affinity.podAffinity.required DuringSchedulingIgnoredDuringExecution[].namespaceS elector.matchExpressions[].values

Description

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

Type

array

.spec.controller.template.pod.affinity.podAffinity.required DuringSchedulingIgnoredDuringExecution[].namespaceS

`elector.matchExpressions[].values[]`

Type

string

`.spec.controller.template.pod.affinity.podAffinity.required DuringSchedulingIgnoredDuringExecution[].namespaceS elector.matchLabels`

Description

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

Type

object

`.spec.controller.template.pod.affinity.podAffinity.required DuringSchedulingIgnoredDuringExecution[].namespaces`

Description

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

Type

array

`.spec.controller.template.pod.affinity.podAffinity.required DuringSchedulingIgnoredDuringExecution[].namespaces[]`

Type

string

.spec.controller.template.pod.affinity.podAntiAffinity

Description

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

Type

object

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

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

`.spec.controller.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

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

Type

array

`.spec.controller.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

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

Type

object

Required

podAffinityTerm

weight

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

.spec.controller.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm

Description

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

Type

object

Required

topologyKey

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

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

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

`.spec.controller.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector`

Description

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

Type

object

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

.spec.controller.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions

Description

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

Type

array

.spec.controller.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]

Description

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

Type

object

Required

key

operator

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

.spec.controller.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

Description

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

Type

array

`.spec.controller.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]`

Type

string

`.spec.controller.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels`

Description

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

Type

object

`.spec.controller.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys`

Description

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

Type

`array`

`.spec.controller.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]`

Type

`string`

`.spec.controller.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys`

Description

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

Type

`array`

`.spec.controller.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]`

Type

`string`

`.spec.controller.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector`

Description

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

Type

object

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

`.spec.controller.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions`

Description

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

Type

array

`.spec.controller.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]`

Description

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

Type

object

Required

key

operator

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

`.spec.controller.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffini`

tyTerm.namespaceSelector.matchExpressions[].values

Description

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

Type

array

.spec.controller.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]

Type

string

.spec.controller.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

Description

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

Type

object

.spec.controller.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

Description

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

Type

array

.spec.controller.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

string

.spec.controller.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

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

Type

array

.spec.controller.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

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

Type

object

Required

`topologyKey`

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

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

Property	Type	Description
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the <code>labelSelector</code> in the specified namespaces, where co-located is defined as running on a node whose value of the label with key <code>topologyKey</code> matches that of any node on which any of the selected pods is running. Empty <code>topologyKey</code> is not allowed.

`.spec.controller.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector`

Description

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

Type

`object`

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

`.spec.controller.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

Description

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

Type

array

`.spec.controller.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]`

Description

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

Type

object

Required

key

operator

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

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

.spec.controller.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values

Description

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

Type

array

.spec.controller.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]

Type

string

.spec.controller.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels

Description

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

Type

object

.spec.controller.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys

Description

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

Type

array

.spec.controller.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]

Type

string

.spec.controller.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatch

LabelKeys

Description

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

Type

array

`.spec.controller.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]`

Type

string

`.spec.controller.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector`

Description

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

Type

object

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

`.spec.controller.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions`

Description

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

Type

`array`

`.spec.controller.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]`

Description

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

Type

`object`

Required

key

operator

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

.spec.controller.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

Description

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

Type

array

.spec.controller.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespace

`eSelector.matchExpressions[].values[]`

Type

string

`.spec.controller.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels`

Description

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

Type

object

`.spec.controller.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces`

Description

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

Type

array

`.spec.controller.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]`

Type

string

.spec.controller.template.pod.securityContext

Description

Security context for pods such as permissions and privilege escalation.

Type

object

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

Property	Type	Description
<p><code>fsGroup</code></p>	<p><code>integer</code></p>	<p>A special supplemental group that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod:</p> <ol style="list-style-type: none"> 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---- <p>If unset, the Kubelet will not modify the ownership and permissions of any volume. Note that this field cannot be set when <code>spec.os.name</code> is windows.</p>
<p><code>fsGroupChangePolicy</code></p>	<p><code>string</code></p>	<p><code>fsGroupChangePolicy</code> defines behavior of changing ownership and permission of the volume before being exposed inside Pod. This field will only apply to volume types which support fsGroup based ownership(and permissions). It will have no effect on ephemeral volume types such as: secret, configmaps and emptydir. Valid values are "OnRootMismatch" and "Always". If not specified, "Always" is used. Note that this field cannot be set when <code>spec.os.name</code> is windows.</p>

Property	Type	Description
<code>runAsGroup</code>	<code>integer</code>	<p>The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.</p>
<code>runAsNonRoot</code>	<code>boolean</code>	<p>Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.</p>
<code>runAsUser</code>	<code>integer</code>	<p>The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.</p>

Property	Type	Description
<code>seLinuxChangePolicy</code>	<code>string</code>	<p><code>seLinuxChangePolicy</code> defines how the container's SELinux label is applied to all volumes used by the Pod. It has no effect on nodes that do not support SELinux or to volumes does not support SELinux. Valid values are "MountOption" and "Recursive".</p> <p>"Recursive" means relabeling of all files on all Pod volumes by the container runtime. This may be slow for large volumes, but allows mixing privileged and unprivileged Pods sharing the same volume on the same node.</p> <p>"MountOption" mounts all eligible Pod volumes with <code>-o context</code> mount option. This requires all Pods that share the same volume to use the same SELinux label. It is not possible to share the same volume among privileged and unprivileged Pods. Eligible volumes are in-tree FibreChannel and iSCSI volumes, and all CSI volumes whose CSI driver announces SELinux support by setting <code>spec.seLinuxMount: true</code> in their CSIDriver instance. Other volumes are always re-labelled recursively.</p> <p>"MountOption" value is allowed only when SELinuxMount feature gate is enabled.</p> <p>If not specified and SELinuxMount feature gate is enabled, "MountOption" is used. If not specified and SELinuxMount feature gate is disabled, "MountOption" is used for ReadWriteOncePod volumes and "Recursive" for all other volumes.</p>

Property	Type	Description
		<p>This field affects only Pods that have SELinux label set, either in PodSecurityContext or in SecurityContext of all containers.</p> <p>All Pods that use the same volume should use the same seLinuxChangePolicy, otherwise some pods can get stuck in ContainerCreating state. Note that this field cannot be set when spec.os.name is windows.</p>
seLinuxOptions	object	<p>The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.</p>
seccompProfile	object	<p>The seccomp options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.</p>
supplementalGroups	array	<p>A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if</p>

Property	Type	Description
		specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.
<code>supplementalGroupsPolicy</code>	<code>string</code>	Defines how supplemental groups of the first container processes are calculated. Valid values are "Merge" and "Strict". If not specified, "Merge" is used. (Alpha) Using the field requires the SupplementalGroupsPolicy feature gate to be enabled and the container runtime must implement support for this feature. Note that this field cannot be set when spec.os.name is windows.
<code>sysctls</code>	<code>array</code>	Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.

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

`.spec.controller.template.pod.securityContext.appArmorProfile`

Description

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

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if <code>type</code> is "Localhost".

Property	Type	Description
type	string	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

.spec.controller.template.pod.securityContext.seLinuxOptions

Description

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

Type

object

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

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

`.spec.controller.template.pod.securityContext.seccompProfile`

Description

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

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if <code>type</code> is "Localhost". Must NOT be set for any other type.

Property	Type	Description
type	string	<p>type indicates which kind of seccomp profile will be applied. Valid options are:</p> <p>Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied.</p>

`.spec.controller.template.pod.securityContext.supplementalGroups`

Description

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

Type

array

`.spec.controller.template.pod.securityContext.supplementalGroups[]`

Type

integer

`.spec.controller.template.pod.securityContext.sysctls`

Description

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

Type

array

`.spec.controller.template.pod.securityContext.sysctls[]`

Description

Sysctl defines a kernel parameter to be set

Type

object

Required

name

value

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

`.spec.controller.template.pod.securityContext.windowsOptions`

Description

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

Type

object

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

Property	Type	Description
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

`.spec.controller.template.pod.tolerations`

Description

Tolerations for pods to schedule onto nodes with taints.

Type

`array`

`.spec.controller.template.pod.tolerations[]`

Description

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

Type

`object`

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

.spec.entityOperator

Description

EntityOperator configuration

Type

object

Property	Type	Description
<code>template</code>	object	
<code>tlsSidecar</code>	object	
<code>topicOperator</code>	object	Configuration for TopicOperator
<code>userOperator</code>	object	Configuration for UserOperator

.spec.entityOperator.template

Type

object

Property	Type	Description
<code>pod</code>	object	Pod template for containers in the pod. Contains security context, affinity and tolerations settings.

.spec.entityOperator.template.pod

Description

Pod template for containers in the pod. Contains security context, affinity and tolerations settings.

Type

object

Property	Type	Description
<code>affinity</code>	object	Affinity and anti-affinity rules for pod scheduling.
<code>enableServiceLinks</code>	boolean	EnableServiceLinks Indicates whether information about services should be injected into Pod's environment variables.
<code>securityContext</code>	object	Security context for pods such as permissions and privilege escalation.
<code>tolerations</code>	array	Tolerations for pods to schedule onto nodes with taints.

`.spec.entityOperator.template.pod.affinity`

Description

Affinity and anti-affinity rules for pod scheduling.

Type

object

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

`.spec.entityOperator.template.pod.affinity.nodeAffinity`

Description

Describes node affinity scheduling rules for the pod.

Type

`object`

Property	Type	Description
<code>preferredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, it may choose a node that violates one or more the expressions. The node that is most

Property	Type	Description
		<p>preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, <code>requiredDuringSchedulingIgnoredDuringExecution</code> affinity expressions, etc.) we compute a sum by iterating through the elements of this field adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest sum are the most preferred.</p>
<p><code>requiredDuringSchedulingIgnoredDuringExecution</code></p>	<p>object</p>	<p>If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a system restart or update), the system may or may not try to eventually evict the pod from its node.</p>

`.spec.entityOperator.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

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

Type

array

`.spec.entityOperator.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

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

Type

object

Required

preference

weight

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

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

`.spec.entityOperator.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference`

Description

A node selector term, associated with the corresponding weight.

Type

`object`

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

`.spec.entityOperator.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions`

Description

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

Type

array

.spec.entityOperator.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[]

Description

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

Type

object

Required

key

operator

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

.spec.entityOperator.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values

Description

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

Type

array

.spec.entityOperator.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values[]

Type

string

.spec.entityOperator.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields

Description

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

Type

array

.spec.entityOperator.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference

nce.matchFields[]

Description

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

Type

object

Required

key

operator

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

.spec.entityOperator.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values

Description

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

Type

array

.spec.entityOperator.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values[]

Type

string

.spec.entityOperator.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

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

Type

object

Required

nodeSelectorTerms

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

`.spec.entityOperator.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms`

Description

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

Type

array

`.spec.entityOperator.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[]`

Description

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

Type

object

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

`.spec.entityOperator.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions`

Description

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

Type

array

.spec.entityOperator.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[]

Description

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

Type

object

Required

key

operator

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

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

`.spec.entityOperator.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values`

Description

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

Type

`array`

`.spec.entityOperator.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values[]`

Type

`string`

`.spec.entityOperator.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSele`

ctorTerms[].matchFields

Description

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

Type

array

.spec.entityOperator.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[]

Description

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

Type

object

Required

key

operator

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

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

.spec.entityOperator.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values

Description

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

Type

array

.spec.entityOperator.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values[]

Type

string

.spec.entityOperator.template.pod.affinity.podAffinity

Description

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

Type

object

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

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

`.spec.entityOperator.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

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

Type

array

`.spec.entityOperator.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

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

Type

object

Required

podAffinityTerm

weight

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

`.spec.entityOperator.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm`

Description

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

Type

object

Required

topologyKey

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

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

`.spec.entityOperator.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector`

Description

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

Type

object

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

`.spec.entityOperator.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions`

Description

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

Type

array

`.spec.entityOperator.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]`

Description

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

Type

object

Required

key

operator

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

`.spec.entityOperator.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values`

Description

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

Type

array

.spec.entityOperator.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]

Type

string

.spec.entityOperator.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels

Description

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

Type

object

.spec.entityOperator.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-

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

Type

array

.spec.entityOperator.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]

Type

string

.spec.entityOperator.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys

Description

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

Type

array

`.spec.entityOperator.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]`

Type

string

`.spec.entityOperator.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector`

Description

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

Type

object

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

.spec.entityOperator.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions

Description

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

Type

array

.spec.entityOperator.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]

Description

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

Type

object

Required

key

operator

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

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

`.spec.entityOperator.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values`

Description

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

Type

`array`

`.spec.entityOperator.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]`

Type

`string`

`.spec.entityOperator.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels`

Description

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

Type

object

.spec.entityOperator.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

Description

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

Type

array

.spec.entityOperator.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

string

.spec.entityOperator.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

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

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

Type

array

.spec.entityOperator.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

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

Type

object

Required

topologyKey

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

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

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

`.spec.entityOperator.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector`

Description

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

Type

object

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

.spec.entityOperator.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions

Description

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

Type

array

.spec.entityOperator.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]

Description

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

Type

object

Required

key

operator

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

.spec.entityOperator.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values

Description

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

Type

array

`.spec.entityOperator.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]`

Type

string

`.spec.entityOperator.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels`

Description

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

Type

object

`.spec.entityOperator.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys`

Description

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

Type

`array`

`.spec.entityOperator.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]`

Type

`string`

`.spec.entityOperator.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys`

Description

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

Type

`array`

`.spec.entityOperator.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]`

Type

`string`

`.spec.entityOperator.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector`

Description

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

Type

object

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

`.spec.entityOperator.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions`

Description

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

Type

array

`.spec.entityOperator.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]`

Description

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

Type

object

Required

key

operator

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

`.spec.entityOperator.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespace`

ceSelector.matchExpressions[].values

Description

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

Type

array

.spec.entityOperator.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]

Type

string

.spec.entityOperator.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels

Description

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

Type

object

.spec.entityOperator.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces

Description

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

Type

array

`.spec.entityOperator.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]`

Type

string

`.spec.entityOperator.template.pod.affinity.podAntiAffinity`

Description

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

Type

object

Property	Type	Description
<code>preferredDuringSchedulingIgnoredDuringExecution</code>	array	The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expression specified by this field, it may choose a node that violates one or more of the expressions. The

Property	Type	Description
		<p>node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution anti-affinity expressions etc.), compute a sum by iterating through the elements of this field adding "weight" to the sum if the node has a podAffinityTerm which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.</p>
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	<p>If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled on the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to pod label update), the system may or may not</p>

Property	Type	Description
		try to eventually evict pod from its node. When there are multiple elements, the lists of nodes corresponding each podAffinityTerm intersected, i.e. all terms must be satisfied.

`.spec.entityOperator.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

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

Type

array

`.spec.entityOperator.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

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

Type

object

Required

podAffinityTerm

weight

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

.spec.entityOperator.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm

Description

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

Type

object

Required

topologyKey

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

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

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

.spec.entityOperator.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector

Description

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

Type

object

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

.spec.entityOperator.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions

Description

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

Type

array

.spec.entityOperator.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]

Description

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

Type

object

Required

key

operator

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

.spec.entityOperator.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

Description

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

Type

array

.spec.entityOperator.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]

Type

string

.spec.entityOperator.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels

Description

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

Type

object

.spec.entityOperator.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys

Description

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

Type

`array`

.spec.entityOperator.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]

Type

`string`

.spec.entityOperator.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys

Description

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

Type

`array`

.spec.entityOperator.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]

Type

`string`

`.spec.entityOperator.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector`

Description

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

Type

object

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

`.spec.entityOperator.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions`

Description

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

Type

array

.spec.entityOperator.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]

Description

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

Type

object

Required

key

operator

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

.spec.entityOperator.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].pod

AffinityTerm.namespaceSelector.matchExpressions[].values

Description

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

Type

array

.spec.entityOperator.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]

Type

string

.spec.entityOperator.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

Description

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

Type

object

.spec.entityOperator.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

Description

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

Type

array

.spec.entityOperator.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

string

.spec.entityOperator.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

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

Type

array

.spec.entityOperator.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

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

Type

object

Required

topologyKey

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

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

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

`.spec.entityOperator.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector`

Description

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

Type

`object`

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

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

`.spec.entityOperator.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

Description

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

Type

array

`.spec.entityOperator.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]`

Description

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

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.

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

`.spec.entityOperator.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labels.elector.matchExpressions[].values`

Description

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

Type

`array`

`.spec.entityOperator.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labels.elector.matchExpressions[].values[]`

Type

`string`

`.spec.entityOperator.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labels.elector.matchLabels`

Description

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

Type

object

`.spec.entityOperator.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys`

Description

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

Type

array

`.spec.entityOperator.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]`

Type

string

`.spec.entityOperator.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys`

Description

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

Type

array

`.spec.entityOperator.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]`

Type

string

`.spec.entityOperator.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field.

null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

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

.spec.entityOperator.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions

Description

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

Type

array

.spec.entityOperator.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]

Description

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

Type

object

Required

key

operator

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

.spec.entityOperator.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

Description

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

Type

array

.spec.entityOperator.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]

Type

string

.spec.entityOperator.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels

Description

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

Type

object

.spec.entityOperator.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces

Description

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

Type

array

`.spec.entityOperator.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]`

Type

string

`.spec.entityOperator.template.pod.securityContext`

Description

Security context for pods such as permissions and privilege escalation.

Type

object

Property	Type	Description
<code>appArmorProfile</code>	object	appArmorProfile is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.
<code>fsGroup</code>	integer	<p>A special supplemental group that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod:</p> <ol style="list-style-type: none"> 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---

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

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

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

Property	Type	Description
		context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
<code>seccompProfile</code>	<code>object</code>	The seccomp options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.
<code>supplementalGroups</code>	<code>array</code>	A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.
<code>supplementalGroupsPolicy</code>	<code>string</code>	Defines how supplemental groups of the first container processes are calculated.

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

`.spec.entityOperator.template.pod.securityContext.appArmorProfile`

Description

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

Type

object

Required

type

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

.spec.entityOperator.template.pod.securityContext.seLinuxOptions

Description

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

Type

object

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

`.spec.entityOperator.template.pod.securityContext.seccompProfile`

Description

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

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	localhostProfile indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured

Property	Type	Description
		seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.
		type indicates which kind of seccomp profile will be applied. Valid options are:
<code>type</code>	<code>string</code>	Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied.

`.spec.entityOperator.template.pod.securityContext.supplementalGroups`

Description

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

Type

`array`

`.spec.entityOperator.template.pod.securityContext.supplementalGroups[]`

Type

`integer`

.spec.entityOperator.template.pod.securityContext.sysctls

Description

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

Type

array

.spec.entityOperator.template.pod.securityContext.sysctls

□

Description

Sysctl defines a kernel parameter to be set

Type

object

Required

name

value

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

.spec.entityOperator.template.pod.securityContext.windowsOptions

Description

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

Type

object

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

Property	Type	Description
		PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

.spec.entityOperator.template.pod.tolerations

Description

Tolerations for pods to schedule onto nodes with taints.

Type

array

.spec.entityOperator.template.pod.tolerations[]

Description

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

Type

object

Property	Type	Description
effect	string	Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.
key	string	Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is

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

.spec.entityOperator.tlsSidecar

Type

`object`

Property	Type	Description
<code>jvmOptions</code>	<code>object</code>	JVM options such as Xms and Xmx

Property	Type	Description
resources	object	Resources defines CPU/memory resource requests/limits

.spec.entityOperator.tlsSidecar.jvmOptions

Description

JVM options such as Xms and Xmx

Type

object

Property	Type	Description
-Xms	string	The Xms field sets the JVM min heap size parameter
-Xmx	string	The Xmx field sets the JVM max heap size parameter

.spec.entityOperator.tlsSidecar.resources

Description

Resources defines CPU/memory resource requests/limits

Type

object

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

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

`.spec.entityOperator.tlsSidecar.resources.claims`

Description

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

Type

array

`.spec.entityOperator.tlsSidecar.resources.claims[]`

Description

ResourceClaim references one entry in PodSpec.ResourceClaims.

Type

object

Required

name

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

.spec.entityOperator.tlsSidecar.resources.limits

Description

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

Type

object

.spec.entityOperator.tlsSidecar.resources.requests

Description

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

Type

object

.spec.entityOperator.topicOperator

Description

Configuration for TopicOperator

Type

object

Property	Type	Description
<code>jvmOptions</code>	object	JVM options such as Xms and Xmx
<code>logging</code>	object	Logging configuration for TopicOperator.
<code>resources</code>	object	Resources defines CPU/memory resource requests/limits

.spec.entityOperator.topicOperator.jvmOptions

Description

JVM options such as Xms and Xmx

Type

object

Property	Type	Description
<code>-Xms</code>	string	The Xms field sets the JVM min heap size parameter

Property	Type	Description
<code>-Xmx</code>	<code>string</code>	The Xmx field sets the JVM max heap size parameter

`.spec.entityOperator.topicOperator.logging`

Description

Logging configuration for TopicOperator.

Type

`object`

Required

`loggers`

`type`

Property	Type	Description
<code>loggers</code>	<code>object</code>	Loggers is a map from logger name to logger level.
<code>type</code>	<code>string</code>	Logging type , it must have the value inline

`.spec.entityOperator.topicOperator.logging.loggers`

Description

Loggers is a map from logger name to logger level.

Type

`object`

`.spec.entityOperator.topicOperator.resources`

Description

Resources defines CPU/memory resource requests/limits

Type

object

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

.spec.entityOperator.topicOperator.resources.claims

Description

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

Type

array

`.spec.entityOperator.topicOperator.resources.claims[]`

Description

ResourceClaim references one entry in `PodSpec.ResourceClaims`.

Type

object

Required

name

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

`.spec.entityOperator.topicOperator.resources.limits`

Description

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

Type

`object`

`.spec.entityOperator.topicOperator.resources.requests`

Description

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

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

Type

`object`

`.spec.entityOperator.userOperator`

Description

Configuration for UserOperator

Type

`object`

Property	Type	Description
<code>jvmOptions</code>	<code>object</code>	JVM options such as Xms and Xmx
<code>logging</code>	<code>object</code>	Logging configuration for UserOperator.
<code>resources</code>	<code>object</code>	Resources defines CPU/memory resource requests/limits

`.spec.entityOperator.userOperator.jvmOptions`

Description

JVM options such as Xms and Xmx

Type

object

Property	Type	Description
-Xms	string	The Xms field sets the JVM min heap size parameter
-Xmx	string	The Xmx field sets the JVM max heap size parameter

.spec.entityOperator.userOperator.logging

Description

Logging configuration for UserOperator.

Type

object

Required

loggers

type

Property	Type	Description
loggers	object	Loggers is a map from logger name to logger level.
type	string	Logging type , it must have the value inline

.spec.entityOperator.userOperator.logging.loggers

Description

Loggers is a map from logger name to logger level.

Type

object

.spec.entityOperator.userOperator.resources

Description

Resources defines CPU/memory resource requests/limits

Type

object

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

Property	Type	Description
		https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/ ↗

.spec.entityOperator.userOperator.resources.claims

Description

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

Type

array

.spec.entityOperator.userOperator.resources.claims[]

Description

ResourceClaim references one entry in `PodSpec.ResourceClaims`.

Type

object

Required

name

Property	Type	Description
name	string	Name must match the name of one entry in <code>pod.spec.resourceClaims</code> 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

Property	Type	Description
		made available, otherwise only the result of this request.

`.spec.entityOperator.userOperator.resources.limits`

Description

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

Type

object

`.spec.entityOperator.userOperator.resources.requests`

Description

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

Type

object

`.spec.kafka`

Description

Kafka configuration

Type

object

Required

listeners

Property	Type	Description
<code>authorization</code>	<code>object</code>	Authorization configuration for brokers
<code>jvmOptions</code>	<code>object</code>	JVM options such as Xms and Xmx for Kafka brokers.
<code>listeners</code>	<code>object</code>	Kafka brokers listeners configuration
<code>logging</code>	<code>object</code>	Logging configuration for Kafka.
<code>template</code>	<code>object</code>	Pod template for brokers

`.spec.kafka.authorization`

Description

Authorization configuration for brokers

Type

`object`

Required

`type`

Property	Type	Description
<code>superUsers</code>	<code>array</code>	SuperUsers field lists the Kafka users that will be super users and have all broker permissions

Property	Type	Description
type	string	

.spec.kafka.authorization.superUsers

Description

SuperUsers field lists the Kafka users that will be super users and have all broker permissions

Type

array

.spec.kafka.authorization.superUsers[]

Type

string

.spec.kafka.jvmOptions

Description

JVM options such as Xms and Xmx for Kafka brokers.

Type

object

Property	Type	Description
-Xms	string	The Xms field sets the JVM min heap size parameter
-Xmx	string	The Xmx field sets the JVM max heap size parameter

.spec.kafka.listeners

Description

Kafka brokers listeners configuration

Type

object

Property	Type	Description
<code>external</code>	object	External listener for routes to bootstrap brokers.
<code>plain</code>	object	Plain listener configuration (unencrypted).
<code>tls</code>	object	TLS listener configuration (encrypted).

.spec.kafka.listeners.external

Description

External listener for routes to bootstrap brokers.

Type

object

Required

`tls` `type`

Property	Type	Description
<code>authentication</code>	object	Authentication field defines the authentication configuration for ExternalListener.

Property	Type	Description
<code>overrides</code>	<code>object</code>	Overrides defines how to override defaults for external brokers and bootstrap broker
<code>tls</code>	<code>boolean</code>	Tls field defines whether to use TLS encryption or not
<code>type</code>	<code>string</code>	Type field defines the connection type

`.spec.kafka.listeners.external.authentication`

Description

Authentication field defines the authentication configuration for ExternalListener.

Type

`object`

Required

`type`

Property	Type	Description
<code>type</code>	<code>string</code>	Type field defines the authentication type for listeners

`.spec.kafka.listeners.external.overrides`

Description

Overrides defines how to override defaults for external brokers and bootstrap broker

Type

object

Property	Type	Description
bootstrap	object	Bootstrap defines configurations specific to the bootstrap svc that override the default configs.
brokers	array	Brokers field defines configurations that override the defaults for each external broker individually.

.spec.kafka.listeners.external.overrides.bootstrap

Description

Bootstrap defines configurations specific to the bootstrap svc that override the default configs.

Type

object

Property	Type	Description
nodePort	integer	NodePort defines the port to expose broker listener

.spec.kafka.listeners.external.overrides.brokers

Description

Brokers field defines configurations that override the defaults for each external broker individually.

Type

array

.spec.kafka.listeners.external.overrides.brokers[]

Type

object

Required

broker

Property	Type	Description
advertisedHost	string	AdvertisedHost is the host reachable from outside the cluster
advertisedPort	integer	AdvertisedPort is the port reachable from outside the cluster
broker	integer	broker ID
nodePort	integer	NodePort exposes the broker port

.spec.kafka.listeners.plain

Description

Plain listener configuration (unencrypted).

Type

object

Property	Type	Description
<code>authentication</code>	<code>object</code>	Authentication field defines the authentication configuration that will be used for PlainListener.

`.spec.kafka.listeners.plain.authentication`

Description

Authentication field defines the authentication configuration that will be used for PlainListener.

Type

`object`

Required

`type`

Property	Type	Description
<code>type</code>	<code>string</code>	Type field defines the authentication type for listeners

`.spec.kafka.listeners.tls`

Description

TLS listener configuration (encrypted).

Type

`object`

Property	Type	Description
<code>authentication</code>	<code>object</code>	Authentication field defines the authentication configuration that will be used for TlsListener.

`.spec.kafka.listeners.tls.authentication`

Description

Authentication field defines the authentication configuration that will be used for TlsListener.

Type

`object`

Required

`type`

Property	Type	Description
<code>type</code>	<code>string</code>	Type field defines the authentication type for listeners

`.spec.kafka.logging`

Description

Logging configuration for Kafka.

Type

`object`

Required

`loggers`

`type`

Property	Type	Description
loggers	object	Loggers is a map from logger name to logger level.
type	string	Logging type , it must have the value inline

.spec.kafka.logging.loggers

Description

Loggers is a map from logger name to logger level.

Type

object

.spec.kafka.template

Description

Pod template for brokers

Type

object

Property	Type	Description
pod	object	Pod template for containers in the pod. Contains security context, affinity and tolerations settings.

.spec.kafka.template.pod

Description

Pod template for containers in the pod. Contains security context, affinity and tolerations settings.

Type

object

Property	Type	Description
<code>affinity</code>	object	Affinity and anti-affinity rules for pod scheduling.
<code>enableServiceLinks</code>	boolean	EnableServiceLinks Indicates whether information about services should be injected into Pod's environment variables.
<code>securityContext</code>	object	Security context for pods such as permissions and privilege escalation.
<code>tolerations</code>	array	Tolerations for pods to schedule onto nodes with taints.

`.spec.kafka.template.pod.affinity`

Description

Affinity and anti-affinity rules for pod scheduling.

Type

object

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

`.spec.kafka.template.pod.affinity.nodeAffinity`

Description

Describes node affinity scheduling rules for the pod.

Type

`object`

Property	Type	Description
<code>preferredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most

Property	Type	Description
		<p>preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, <code>requiredDuringSchedulingIgnoredDuringExecution</code> affinity expressions, etc.) we compute a sum by iterating through the elements of this field adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest sum are the most preferred.</p>
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>object</code>	<p>If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a system restart or update), the system may or may not try to eventually evict the pod from its node.</p>

.spec.kafka.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

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

Type

array

.spec.kafka.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

Description

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

Type

object

Required

preference

weight

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

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

`.spec.kafka.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference`

Description

A node selector term, associated with the corresponding weight.

Type

`object`

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

`.spec.kafka.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions`

Description

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

Type

`array`

`.spec.kafka.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchesExpressions[]`

Description

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

Type

object

Required

key

operator

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

`.spec.kafka.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchesExpressions[]`

hExpressions[].values

Description

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

Type

array

.spec.kafka.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchesExpressions[].values[]

Type

string

.spec.kafka.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchesFields

Description

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

Type

array

.spec.kafka.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchesFields[]

Description

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

Type

object

Required

key

operator

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

`.spec.kafka.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matches[].values`

Description

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

Type

array

.spec.kafka.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values[]

Type

string

.spec.kafka.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

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

Type

object

Required

nodeSelectorTerms

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

.spec.kafka.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerm

S

Description

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

Type

array

`.spec.kafka.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[]`

Description

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

Type

object

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

`.spec.kafka.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions`

Description

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

Type

array

`.spec.kafka.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[]`

Description

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

Type

object

Required

key

operator

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

.spec.kafka.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values

Description

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

Type

array

.spec.kafka.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values[]

Type

string

.spec.kafka.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields

Description

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

Type

array

.spec.kafka.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerm

s[].matchFields[]

Description

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

Type

object

Required

key

operator

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

.spec.kafka.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values

Description

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

Type

array

`.spec.kafka.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values[]`

Type

string

`.spec.kafka.template.pod.affinity.podAffinity`

Description

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

Type

object

Property	Type	Description
<code>preferredDuringSchedulingIgnoredDuringExecution</code>	array	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most

Property	Type	Description
		<p>preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution affinity expressions, etc.) we compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has a podAffinityTerm which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.</p>
<p><code>requiredDuringSchedulingIgnoredDuringExecution</code></p>	<p><code>array</code></p>	<p>If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to pod label update), the system may or may not try to eventually evict the pod from its node. When</p>

Property	Type	Description
		there are multiple elements, the lists of nodes corresponding each podAffinityTerm intersected, i.e. all terms must be satisfied.

`.spec.kafka.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

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

Type

array

`.spec.kafka.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

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

Type

object

Required

podAffinityTerm

weight

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

.spec.kafka.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm**Description**

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

Type

object

Required

topologyKey

Property	Type	Description
labelSelector	object	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
matchLabelKeys	array	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The

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

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

`.spec.kafka.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector`

Description

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

Type

object

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

.spec.kafka.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions

Description

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

Type

array

.spec.kafka.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]

Description

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

Type

object

Required

key

operator

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

.spec.kafka.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

Description

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

Type

array

.spec.kafka.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.l

abelSelector.matchExpressions[].values[]

Type

string

.spec.kafka.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels

Description

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

Type

object

.spec.kafka.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys

Description

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

Type

array

`.spec.kafka.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]`

Type

string

`.spec.kafka.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys`

Description

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

Type

array

`.spec.kafka.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]`

Type

string

`.spec.kafka.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespacesSelector`

Description

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

Type

object

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

`.spec.kafka.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespacesSelector.matchExpressions`

Description

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

Type

array

`.spec.kafka.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces.selector.matchExpressions[]`

Description

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

Type

object

Required

key

operator

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

`.spec.kafka.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces.selector.matchExpressions[]`

namespaceSelector.matchExpressions[].values

Description

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

Type

array

.spec.kafka.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]

Type

string

.spec.kafka.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

Description

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

Type

object

.spec.kafka.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

Description

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

Type

array

.spec.kafka.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

string

.spec.kafka.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

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

Type

array

.spec.kafka.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

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

Type

object

Required

`topologyKey`

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

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

Property	Type	Description
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the <code>labelSelector</code> in the specified namespaces, where co-located is defined as running on a node whose value of the label with key <code>topologyKey</code> matches that of any node on which any of the selected pods is running. Empty <code>topologyKey</code> is not allowed.

`.spec.kafka.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector`

Description

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

Type

`object`

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

.spec.kafka.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions

Description

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

Type

array

.spec.kafka.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]

Description

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

Type

object

Required

key

operator

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

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

.spec.kafka.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values

Description

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

Type

array

.spec.kafka.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]

Type

string

.spec.kafka.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels

Description

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

Type

object

`.spec.kafka.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys`

Description

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

Type

array

`.spec.kafka.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]`

Type

string

`.spec.kafka.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys`

S

Description

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

Type

array

.spec.kafka.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]

Type

string

.spec.kafka.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector

Description

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

Type

object

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

`.spec.kafka.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions`

Description

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

Type

`array`

`.spec.kafka.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]`

Description

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

Type

`object`

Required

key operator

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

.spec.kafka.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

Description

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

Type

array

.spec.kafka.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector

`r.matchExpressions[].values[]`

Type

string

`.spec.kafka.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels`

Description

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

Type

object

`.spec.kafka.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces`

Description

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

Type

array

`.spec.kafka.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]`

Type

string

.spec.kafka.template.pod.affinity.podAntiAffinity

Description

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

Type

object

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

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

`.spec.kafka.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

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

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

Type

array

.spec.kafka.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

Description

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

Type

object

Required

podAffinityTerm

weight

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

.spec.kafka.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTe

rm

Description

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

Type

object

Required

topologyKey

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

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

Property	Type	Description
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.kafka.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector`

Description

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

Type

`object`

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

`.spec.kafka.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions`

Description

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

Type

array

`.spec.kafka.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]`

Description

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

Type

object

Required

key

operator

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

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

`.spec.kafka.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values`

Description

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

Type

`array`

`.spec.kafka.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]`

Type

`string`

`.spec.kafka.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels`

Description

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

Type

object

.spec.kafka.template.pod.affinity.podAntiAffinity.preferred DuringSchedulingIgnoredDuringExecution[].podAffinityTe rm.matchLabelKeys

Description

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

Type

array

.spec.kafka.template.pod.affinity.podAntiAffinity.preferred DuringSchedulingIgnoredDuringExecution[].podAffinityTe rm.matchLabelKeys[]

Type

string

.spec.kafka.template.pod.affinity.podAntiAffinity.preferred DuringSchedulingIgnoredDuringExecution[].podAffinityTe

rm.mismatchLabelKeys

Description

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

Type

array

.spec.kafka.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]

Type

string

.spec.kafka.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector

Description

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

Type

object

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

`.spec.kafka.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions`

Description

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

Type

`array`

`.spec.kafka.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]`

Description

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

Type

`object`

Required

key

operator

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

.spec.kafka.template.pod.affinity.podAntiAffinity.preferred DuringSchedulingIgnoredDuringExecution[].podAffinityTe rm.namespaceSelector.matchExpressions[].values

Description

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

Type

array

.spec.kafka.template.pod.affinity.podAntiAffinity.preferred DuringSchedulingIgnoredDuringExecution[].podAffinityTe

rm.namespaceSelector.matchExpressions[].values[]

Type

string

.spec.kafka.template.pod.affinity.podAntiAffinity.preferred DuringSchedulingIgnoredDuringExecution[].podAffinityTe rm.namespaceSelector.matchLabels

Description

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

Type

object

.spec.kafka.template.pod.affinity.podAntiAffinity.preferred DuringSchedulingIgnoredDuringExecution[].podAffinityTe rm.namespaces

Description

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

Type

array

.spec.kafka.template.pod.affinity.podAntiAffinity.preferred DuringSchedulingIgnoredDuringExecution[].podAffinityTe rm.namespaces[]

Type

string

.spec.kafka.template.pod.affinity.podAntiAffinity.required DuringSchedulingIgnoredDuringExecution

Description

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

Type

array

.spec.kafka.template.pod.affinity.podAntiAffinity.required DuringSchedulingIgnoredDuringExecution[]

Description

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

Type

object

Required

topologyKey

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

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

`.spec.kafka.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector`

Description

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

Type

object

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

`.spec.kafka.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

Description

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

Type

array

`.spec.kafka.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]`

Description

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

Type

object

Required

key

operator

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

`.spec.kafka.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values`

Description

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

Type

array

**.spec.kafka.template.pod.affinity.podAntiAffinity.required
DuringSchedulingIgnoredDuringExecution[].labelSelector.
matchExpressions[].values[]**

Type

string

**.spec.kafka.template.pod.affinity.podAntiAffinity.required
DuringSchedulingIgnoredDuringExecution[].labelSelector.
matchLabels**

Description

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

Type

object

**.spec.kafka.template.pod.affinity.podAntiAffinity.required
DuringSchedulingIgnoredDuringExecution[].matchLabelK
eys**

Description

MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-

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

Type

array

.spec.kafka.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]

Type

string

.spec.kafka.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys

Description

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

Type

array

`.spec.kafka.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]`

Type

string

`.spec.kafka.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector`

Description

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

Type

object

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

`.spec.kafka.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions`

Description

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

Type

array

`.spec.kafka.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]`

Description

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

Type

object

Required

key

operator

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

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

.spec.kafka.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

Description

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

Type

array

.spec.kafka.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]

Type

string

.spec.kafka.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels

Description

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

Type

object

.spec.kafka.template.pod.affinity.podAntiAffinity.required DuringSchedulingIgnoredDuringExecution[].namespaces

Description

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

Type

array

.spec.kafka.template.pod.affinity.podAntiAffinity.required DuringSchedulingIgnoredDuringExecution[].namespaces[]

Type

string

.spec.kafka.template.pod.securityContext

Description

Security context for pods such as permissions and privilege escalation.

Type

object

Property	Type	Description
<code>appArmorProfile</code>	<code>object</code>	<p><code>appArmorProfile</code> is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when <code>spec.os.name</code> is <code>windows</code>.</p>
<code>fsGroup</code>	<code>integer</code>	<p>A special supplemental group that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod:</p> <ol style="list-style-type: none">1. The owning GID will be the <code>FSGroup</code>2. The <code>setgid</code> bit is set (new files created in the volume will be owned by <code>FSGroup</code>)3. The permission bits are OR'd with <code>rw-rw---</code> <p>If unset, the Kubelet will not modify the ownership and permissions of any volume. Note that this field cannot be set when <code>spec.os.name</code> is <code>windows</code>.</p>
<code>fsGroupChangePolicy</code>	<code>string</code>	<p><code>fsGroupChangePolicy</code> defines behavior of changing ownership and permission of the volume before being exposed inside Pod. This field will only apply to volume types which support <code>fsGroup</code> based ownership(and permissions). It will have no effect on ephemeral volume types such as: <code>secret</code>, <code>configmaps</code> and <code>emptydir</code>. Valid values are <code>"OnRootMismatch"</code> and</p>

Property	Type	Description
		"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

Property	Type	Description
		container. Note that this field cannot be set when spec.os.name is windows.
<code>seLinuxChangePolicy</code>	<code>string</code>	<p><code>seLinuxChangePolicy</code> defines how the container's SELinux label is applied to all volumes used by the Pod. It has no effect on nodes that do not support SELinux or to volumes does not support SELinux. Valid values are "MountOption" and "Recursive".</p> <p>"Recursive" means relabeling of all files on all Pod volumes by the container runtime. This may be slow for large volumes, but allows mixing privileged and unprivileged Pods sharing the same volume on the same node.</p> <p>"MountOption" mounts all eligible Pod volumes with <code>-o context</code> mount option. This requires all Pods that share the same volume to use the same SELinux label. It is not possible to share the same volume among privileged and unprivileged Pods. Eligible volumes are in-tree FibreChannel and iSCSI volumes, and all CSI volumes whose CSI driver announces SELinux support by setting <code>spec.seLinuxMount: true</code> in their CSIDriver instance. Other volumes are always re-labelled recursively.</p> <p>"MountOption" value is allowed only when SELinuxMount feature gate is enabled.</p> <p>If not specified and SELinuxMount feature gate is enabled, "MountOption" is used. If not specified and SELinuxMount feature</p>

Property	Type	Description
		<p>gate is disabled, "MountOption" is used for ReadWriteOncePod volumes and "Recursive" for all other volumes.</p> <p>This field affects only Pods that have SELinux label set, either in PodSecurityContext or in SecurityContext of all containers.</p> <p>All Pods that use the same volume should use the same seLinuxChangePolicy, otherwise some pods can get stuck in ContainerCreating state. Note that this field cannot be set when spec.os.name is windows.</p>
<p>seLinuxOptions</p>	<p>object</p>	<p>The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.</p>
<p>seccompProfile</p>	<p>object</p>	<p>The seccomp options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.</p>

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

Property	Type	Description
		to launch. Note that this field cannot be set when spec.os.name is windows.
<code>windowsOptions</code>	<code>object</code>	The Windows specific settings applied to all containers. If unspecified, the options within a container's SecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

`.spec.kafka.template.pod.securityContext.appArmorProfile`

Description

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

Type

`object`

Required

`type`

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

`.spec.kafka.template.pod.securityContext.seLinuxOptions`

Description

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

Type

`object`

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

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

`.spec.kafka.template.pod.securityContext.seccompProfile`

Description

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

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if <code>type</code> is "Localhost". Must NOT be set for any other type.

Property	Type	Description
<code>type</code>	<code>string</code>	<p>type indicates which kind of seccomp profile will be applied. Valid options are:</p> <p>Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied.</p>

`.spec.kafka.template.pod.securityContext.supplementalGroups`

Description

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

Type

`array`

`.spec.kafka.template.pod.securityContext.supplementalGroups[]`

Type

`integer`

`.spec.kafka.template.pod.securityContext.sysctls`

Description

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

Type

array

`.spec.kafka.template.pod.securityContext.sysctls[]`

Description

Sysctl defines a kernel parameter to be set

Type

object

Required

name

value

Property	Type	Description
<code>name</code>	<code>string</code>	Name of a property to set
<code>value</code>	<code>string</code>	Value of a property to set

`.spec.kafka.template.pod.securityContext.windowsOption`

S

Description

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

Type

object

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

Property	Type	Description
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

`.spec.kafka.template.pod.tolerations`

Description

Tolerations for pods to schedule onto nodes with taints.

Type

`array`

`.spec.kafka.template.pod.tolerations[]`

Description

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

Type

`object`

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

.spec.kafkaExporter

Description

Kafka exporter configuration

Type

object

Property	Type	Description
<code>groupRegex</code>	<code>string</code>	The GroupRegex field specifies regex for scraping metrics from consumer groups
<code>resources</code>	<code>object</code>	Resources defines CPU/memory resource requests/limits
<code>template</code>	<code>object</code>	The Template field customizes the pod configuration by applying a PodTemplate
<code>topicRegex</code>	<code>string</code>	The TopicRegex field specifies regex for scraping metrics from topics

.spec.kafkaExporter.resources

Description

Resources defines CPU/memory resource requests/limits

Type

object

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

`.spec.kafkaExporter.resources.claims`

Description

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

Type

`array`

.spec.kafkaExporter.resources.claims[]

Description

ResourceClaim references one entry in PodSpec.ResourceClaims.

Type

object

Required

name

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

.spec.kafkaExporter.resources.limits

Description

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

Type

object

.spec.kafkaExporter.resources.requests

Description

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

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

Type

object

.spec.kafkaExporter.template

Description

The Template field customizes the pod configuration by applying a PodTemplate

Type

object

Property	Type	Description
pod	object	Pod template for containers in the pod. Contains security context, affinity and tolerations settings.

.spec.kafkaExporter.template.pod

Description

Pod template for containers in the pod. Contains security context, affinity and tolerations settings.

Type

object

Property	Type	Description
affinity	object	Affinity and anti-affinity rules for pod scheduling.

Property	Type	Description
<code>enableServiceLinks</code>	<code>boolean</code>	EnableServiceLinks Indicates whether information about services should be injected into Pod's environment variables.
<code>securityContext</code>	<code>object</code>	Security context for pods such as permissions and privilege escalation.
<code>tolerations</code>	<code>array</code>	Tolerations for pods to schedule onto nodes with taints.

`.spec.kafkaExporter.template.pod.affinity`

Description

Affinity and anti-affinity rules for pod scheduling.

Type

`object`

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

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

`.spec.kafkaExporter.template.pod.affinity.nodeAffinity`

Description

Describes node affinity scheduling rules for the pod.

Type

`object`

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

Property	Type	Description
		iterating through the elements of this field adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest sum are the most preferred.
<code>requiredDuringSchedulingIgnoredDuringExecution</code>	<code>object</code>	<p>If the affinity requirement specified by this field is not met at scheduling time, the pod will not be scheduled onto the node.</p> <p>If the affinity requirement specified by this field ceases to be met at some point during pod execution (e.g. due to a node update), the system may or may not try to eventually evict the pod from its node.</p>

`.spec.kafkaExporter.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights,

i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest sum are the most preferred.

Type

array

.spec.kafkaExporter.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

Description

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

Type

object

Required

preference

weight

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

.spec.kafkaExporter.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference

Description

A node selector term, associated with the corresponding weight.

Type

object

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

`.spec.kafkaExporter.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions`

Description

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

Type

array

`.spec.kafkaExporter.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[]`

Description

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

Type

object

Required

key operator

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

.spec.kafkaExporter.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values

Description

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

Type

array

.spec.kafkaExporter.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values[]

Type

string

.spec.kafkaExporter.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields

Description

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

Type

array

.spec.kafkaExporter.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[]

Description

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

Type

object

Required

key

operator

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

.spec.kafkaExporter.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values

Description

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

Type

array

.spec.kafkaExporter.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference

ce.matchFields[].values[]

Type

string

.spec.kafkaExporter.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

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

Type

object

Required

nodeSelectorTerms

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

.spec.kafkaExporter.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms

Description

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

Type

`array`

`.spec.kafkaExporter.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[]`

Description

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

Type

`object`

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

`.spec.kafkaExporter.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions`

Description

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

Type

`array`

`.spec.kafkaExporter.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[]`

Description

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

Type

object

Required

key

operator

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

`.spec.kafkaExporter.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSele`

ctorTerms[].matchExpressions[].values

Description

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

Type

array

.spec.kafkaExporter.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values[]

Type

string

.spec.kafkaExporter.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields

Description

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

Type

array

.spec.kafkaExporter.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[]

Description

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

Type

object

Required

key

operator

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

.spec.kafkaExporter.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values

Description

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

Type

array

.spec.kafkaExporter.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values[]

Type

string

.spec.kafkaExporter.template.pod.affinity.podAffinity

Description

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

Type

object

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

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

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

`.spec.kafkaExporter.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

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

Type

array

`.spec.kafkaExporter.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

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

Type

object

Required

`podAffinityTerm`

`weight`

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

`.spec.kafkaExporter.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm`

Description

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

Type

object

Required

topologyKey

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

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

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

`.spec.kafkaExporter.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector`

Description

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

Type

object

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

`.spec.kafkaExporter.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions`

Description

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

Type

`array`

`.spec.kafkaExporter.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]`

Description

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

Type

`object`

Required

key operator

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

.spec.kafkaExporter.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

Description

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

Type

array

.spec.kafkaExporter.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffini

tyTerm.labelSelector.matchExpressions[].values[]

Type

string

.spec.kafkaExporter.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinity.terms[].labelSelector.matchLabels

Description

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

Type

object

.spec.kafkaExporter.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinity.terms[].matchLabelKeys

Description

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

Type

array

`.spec.kafkaExporter.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]`

Type

string

`.spec.kafkaExporter.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys`

Description

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

Type

array

`.spec.kafkaExporter.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]`

Type

string

`.spec.kafkaExporter.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector`

Description

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

Type

object

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

`.spec.kafkaExporter.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions`

Description

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

Type

array

`.spec.kafkaExporter.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]`

Description

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

Type

object

Required

key

operator

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

`.spec.kafkaExporter.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffini`

tyTerm.namespaceSelector.matchExpressions[].values

Description

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

Type

array

.spec.kafkaExporter.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinity.terms[].namespaceSelector.matchExpressions[].values[]

Type

string

.spec.kafkaExporter.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinity.terms[].namespaceSelector.matchLabels

Description

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

Type

object

.spec.kafkaExporter.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinity.terms[].namespaces

Description

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

Type

array

.spec.kafkaExporter.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

string

.spec.kafkaExporter.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

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

Type

array

.spec.kafkaExporter.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

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

Type

object

Required

`topologyKey`

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

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

Property	Type	Description
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the <code>labelSelector</code> in the specified namespaces, where co-located is defined as running on a node whose value of the label with key <code>topologyKey</code> matches that of any node on which any of the selected pods is running. Empty <code>topologyKey</code> is not allowed.

`.spec.kafkaExporter.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector`

Description

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

Type

`object`

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

`.spec.kafkaExporter.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

Description

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

Type

array

`.spec.kafkaExporter.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]`

Description

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

Type

object

Required

key

operator

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

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

.spec.kafkaExporter.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values

Description

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

Type

array

.spec.kafkaExporter.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]

Type

string

.spec.kafkaExporter.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels

Description

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

Type

object

.spec.kafkaExporter.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys

Description

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

Type

array

.spec.kafkaExporter.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]

Type

string

.spec.kafkaExporter.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatch

LabelKeys

Description

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

Type

array

`.spec.kafkaExporter.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]`

Type

string

`.spec.kafkaExporter.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector`

Description

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

Type

object

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

`.spec.kafkaExporter.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions`

Description

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

Type

array

`.spec.kafkaExporter.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]`

Description

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

Type

object

Required

key operator

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

.spec.kafkaExporter.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

Description

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

Type

array

.spec.kafkaExporter.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespace

`eSelector.matchExpressions[].values[]`

Type

string

`.spec.kafkaExporter.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels`

Description

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

Type

object

`.spec.kafkaExporter.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces`

Description

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

Type

array

`.spec.kafkaExporter.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]`

Type

string

.spec.kafkaExporter.template.pod.affinity.podAntiAffinity

Description

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

Type

object

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

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

`.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

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

Type

array

.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.pREFERRED_DURING_SCHEDULING_IGNORED_DURING_EXECUTION[]

Description

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

Type

object

Required

podAffinityTerm

weight

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

.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.podAffinityTerm

Description

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

Type

object

Required

topologyKey

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

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

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

`.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector`

Description

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

Type

object

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

.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.p referredDuringSchedulingIgnoredDuringExecution[].podA ffinityTerm.labelSelector.matchExpressions

Description

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

Type

array

.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.p referredDuringSchedulingIgnoredDuringExecution[].podA ffinityTerm.labelSelector.matchExpressions[]

Description

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

Type

object

Required

key

operator

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

.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.pREFERRED_DURING_SCHEDULING_IGNORED_DURING_EXECUTION[].podAffinityTerm.labelSelector.matchExpressions[].values

Description

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

Type

array

**.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.p
referredDuringSchedulingIgnoredDuringExecution[].podA
ffinityTerm.labelSelector.matchExpressions[].values[]**

Type

string

**.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.p
referredDuringSchedulingIgnoredDuringExecution[].podA
ffinityTerm.labelSelector.matchLabels**

Description

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

Type

object

**.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.p
referredDuringSchedulingIgnoredDuringExecution[].podA
ffinityTerm.matchLabelKeys**

Description

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

Type

`array`

`.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]`

Type

`string`

`.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys`

Description

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

Type

`array`

`.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]`

Type

`string`

.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.podAffinityTerm.namespaceSelector.matchExpressions

Description

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

Type

object

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

.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.podAffinityTerm.namespaceSelector.matchExpressions

Description

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

Type

array

**.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.p
referredDuringSchedulingIgnoredDuringExecution[].podA
ffinityTerm.namespaceSelector.matchExpressions[]**

Description

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

Type

object

Required

key

operator

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

**.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.p
referredDuringSchedulingIgnoredDuringExecution[].podA**

ffinityTerm.namespaceSelector.matchExpressions[].value

S

Description

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

Type

array

.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.p referredDuringSchedulingIgnoredDuringExecution[].podA ffinityTerm.namespaceSelector.matchExpressions[].value s[]

Type

string

.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.p referredDuringSchedulingIgnoredDuringExecution[].podA ffinityTerm.namespaceSelector.matchLabels

Description

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

Type

object

`.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces`

Description

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

Type

array

`.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]`

Type

string

`.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution`

Description

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

Type

array

.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

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

Type

object

Required

topologyKey

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

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

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

`.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector`

Description

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

Type

`object`

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

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

`.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

Description

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

Type

array

`.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]`

Description

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

Type

object

Required

key

operator

Property	Type	Description
key	string	key is the label key that the selector applies to.

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

`.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labels.elector.matchExpressions[].values`

Description

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

Type

`array`

`.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labels.elector.matchExpressions[].values[]`

Type

`string`

`.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labels.elector.matchLabels`

Description

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

Type

object

`.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys`

Description

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

Type

array

`.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]`

Type

`string`

`.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys`

Description

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

Type

`array`

`.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]`

Type

`string`

`.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector`

Description

A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the `namespaces` field.

null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.

Type

object

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

.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions

Description

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

Type

array

.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]

Description

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

Type

object

Required

key

operator

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

`.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values`

Description

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

Type

array

.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]

Type

string

.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels

Description

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

Type

object

.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces

Description

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

Type

array

`.spec.kafkaExporter.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]`

Type

string

`.spec.kafkaExporter.template.pod.securityContext`

Description

Security context for pods such as permissions and privilege escalation.

Type

object

Property	Type	Description
<code>appArmorProfile</code>	object	appArmorProfile is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.
<code>fsGroup</code>	integer	<p>A special supplemental group that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod:</p> <ol style="list-style-type: none"> 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- ---

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

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

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

Property	Type	Description
		context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
<code>seccompProfile</code>	<code>object</code>	The seccomp options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.
<code>supplementalGroups</code>	<code>array</code>	A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.
<code>supplementalGroupsPolicy</code>	<code>string</code>	Defines how supplemental groups of the first container processes are calculated.

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

`.spec.kafkaExporter.template.pod.securityContext.appArmorProfile`

Description

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

Type

object

Required

type

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

.spec.kafkaExporter.template.pod.securityContext.seLinuxOptions**Description**

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

Type

object

Property	Type	Description
level	string	Level is SELinux level label that applies to the container.
role	string	Role is a SELinux role label that applies to the container.
type	string	Type is a SELinux type label that applies to the container.
user	string	User is a SELinux user label that applies to the container.

`.spec.kafkaExporter.template.pod.securityContext.seccompProfile`

Description

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

Type

object

Required

type

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured

Property	Type	Description
		seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.
		type indicates which kind of seccomp profile will be applied. Valid options are:
<code>type</code>	<code>string</code>	Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied.

`.spec.kafkaExporter.template.pod.securityContext.supplementalGroups`

Description

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

Type

`array`

`.spec.kafkaExporter.template.pod.securityContext.supplementalGroups[]`

Type

`integer`

`.spec.kafkaExporter.template.pod.securityContext.sysctls`

Description

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

Type

array

`.spec.kafkaExporter.template.pod.securityContext.sysctls[]`

Description

Sysctl defines a kernel parameter to be set

Type

object

Required

name

value

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

`.spec.kafkaExporter.template.pod.securityContext.window sOptions`

Description

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

Type

object

Property	Type	Description
<code>gmsaCredentialSpec</code>	string	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa ^) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
<code>gmsaCredentialSpecName</code>	string	<code>GMSACredentialSpecName</code> is the name of the GMSA credential spec to use.
<code>hostProcess</code>	boolean	<code>HostProcess</code> determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective <code>HostProcess</code> value (it is not allowed to have a mix of <code>HostProcess</code> containers and non- <code>HostProcess</code> containers). In addition, if <code>HostProcess</code> is true then <code>HostNetwork</code> must also be set to true.
<code>runAsUserName</code>	string	The <code>UserName</code> 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

Property	Type	Description
		PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

`.spec.kafkaExporter.template.pod.tolerations`

Description

Tolerations for pods to schedule onto nodes with taints.

Type

array

`.spec.kafkaExporter.template.pod.tolerations[]`

Description

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

Type

object

Property	Type	Description
<code>effect</code>	<code>string</code>	Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.
<code>key</code>	<code>string</code>	Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is

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

.spec.resources

Description

Resource requests/limits for pods

Type

`object`

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

`.spec.resources.claims`

Description

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

Type

`array`

.spec.resources.claims[]

Description

ResourceClaim references one entry in PodSpec.ResourceClaims.

Type

object

Required

name

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

.spec.resources.limits

Description

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

Type

object

.spec.resources.requests

Description

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

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

Type

object

.spec.storage

Description

Storage configuration for persistent volumes

Type

object

Required

class

size

Property	Type	Description
class	string	
deleteClaim	boolean	
size		The name of the StorageClass to claim a PersistentVolume from.

.spec.upgradeOption

Description

UpgradeOption defines the upgrade strategy for the Kafka instance.

Type

object

Property	Type	Description
autoUpgrade	boolean	AutoUpgrade is the flag to auto upgrade the MySQL
crVersion	string	CRVersion is the version of the CR

.spec.zookeeper

Description

Zookeeper configuration

Type

object

Required

replicas

resources

storage

Property	Type	Description
jvmOptions	object	JvmOptions defines Java options for the Zookeeper process
logging	object	Logging configuration for Zookeeper.
replicas	integer	Replicas defines the number of Zookeeper pods
resources	object	Resources defines CPU/memory resource requests/limits

Property	Type	Description
<code>storage</code>	<code>object</code>	Storage defines the volumes used for data persistence
<code>template</code>	<code>object</code>	Template customizes pods using a PodTemplate

`.spec.zookeeper.jvmOptions`

Description

JvmOptions defines Java options for the Zookeeper process

Type

`object`

Property	Type	Description
<code>-Xms</code>	<code>string</code>	The Xms field sets the JVM min heap size parameter
<code>-Xmx</code>	<code>string</code>	The Xmx field sets the JVM max heap size parameter

`.spec.zookeeper.logging`

Description

Logging configuration for Zookeeper.

Type

`object`

Required

`loggers` `type`

Property	Type	Description
loggers	object	Loggers is a map from logger name to logger level.
type	string	Logging type , it must have the value inline

.spec.zookeeper.logging.loggers

Description

Loggers is a map from logger name to logger level.

Type

object

.spec.zookeeper.resources

Description

Resources defines CPU/memory resource requests/limits

Type

object

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

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

.spec.zookeeper.resources.claims

Description

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

Type

array

.spec.zookeeper.resources.claims[]

Description

ResourceClaim references one entry in PodSpec.ResourceClaims.

Type

object

Required

name

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

.spec.zookeeper.resources.limits

Description

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

Type

object

.spec.zookeeper.resources.requests

Description

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

Type

object

.spec.zookeeper.storage

Description

Storage defines the volumes used for data persistence

Type

object

Required

class

size

Property	Type	Description
class	string	
deleteClaim	boolean	
size		The name of the StorageClass to claim a PersistentVolume from.

.spec.zookeeper.template

Description

Template customizes pods using a PodTemplate

Type

object

Property	Type	Description
pod	object	Pod template for containers in the pod. Contains security context, affinity and tolerations settings.

.spec.zookeeper.template.pod

Description

Pod template for containers in the pod. Contains security context, affinity and tolerations settings.

Type

object

Property	Type	Description
<code>affinity</code>	object	Affinity and anti-affinity rules for pod scheduling.
<code>enableServiceLinks</code>	boolean	EnableServiceLinks Indicates whether information about services should be injected into Pod's environment variables.
<code>securityContext</code>	object	Security context for pods such as permissions and privilege escalation.
<code>tolerations</code>	array	Tolerations for pods to schedule onto nodes with taints.

`.spec.zookeeper.template.pod.affinity`

Description

Affinity and anti-affinity rules for pod scheduling.

Type

object

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

`.spec.zookeeper.template.pod.affinity.nodeAffinity`

Description

Describes node affinity scheduling rules for the pod.

Type

`object`

Property	Type	Description
<code>preferredDuringSchedulingIgnoredDuringExecution</code>	<code>array</code>	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most

Property	Type	Description
		<p>preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, <code>requiredDuringSchedulingIgnoredDuringExecution</code> affinity expressions, etc.) we compute a sum by iterating through the elements of this field adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest sum are the most preferred.</p>
<p><code>requiredDuringSchedulingIgnoredDuringExecution</code></p>	<p>object</p>	<p>If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a system restart or update), the system may or may not try to eventually evict the pod from its node.</p>

`.spec.zookeeper.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

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

Type

array

`.spec.zookeeper.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

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

Type

object

Required

preference

weight

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

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

`.spec.zookeeper.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference`

Description

A node selector term, associated with the corresponding weight.

Type

`object`

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

`.spec.zookeeper.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions`

Description

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

Type

`array`

`.spec.zookeeper.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[]`

Description

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

Type

object

Required

key

operator

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

`.spec.zookeeper.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.`

matchExpressions[].values

Description

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

Type

array

.spec.zookeeper.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values[]

Type

string

.spec.zookeeper.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields

Description

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

Type

array

.spec.zookeeper.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[]

Description

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

Type

object

Required

key

operator

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

.spec.zookeeper.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values

Description

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

Type

array

.spec.zookeeper.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values[]

Type

string

.spec.zookeeper.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

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

Type

object

Required

nodeSelectorTerms

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

.spec.zookeeper.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelector

Terms

Description

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

Type

array

`.spec.zookeeper.template.pod.affinity.nodeAffinity.require dDuringSchedulingIgnoredDuringExecution.nodeSelector Terms[]`

Description

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

Type

object

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

`.spec.zookeeper.template.pod.affinity.nodeAffinity.require dDuringSchedulingIgnoredDuringExecution.nodeSelector Terms[].matchExpressions`

Description

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

Type

array

.spec.zookeeper.template.pod.affinity.nodeAffinity.require dDuringSchedulingIgnoredDuringExecution.nodeSelector Terms[].matchExpressions[]

Description

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

Type

object

Required

key

operator

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

.spec.zookeeper.template.pod.affinity.nodeAffinity.require dDuringSchedulingIgnoredDuringExecution.nodeSelector Terms[].matchExpressions[].values

Description

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

Type

array

.spec.zookeeper.template.pod.affinity.nodeAffinity.require dDuringSchedulingIgnoredDuringExecution.nodeSelector Terms[].matchExpressions[].values[]

Type

string

.spec.zookeeper.template.pod.affinity.nodeAffinity.require dDuringSchedulingIgnoredDuringExecution.nodeSelector Terms[].matchFields

Description

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

Type

array

.spec.zookeeper.template.pod.affinity.nodeAffinity.require dDuringSchedulingIgnoredDuringExecution.nodeSelector

Terms[].matchFields[]

Description

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

Type

object

Required

key

operator

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

.spec.zookeeper.template.pod.affinity.nodeAffinity.require dDuringSchedulingIgnoredDuringExecution.nodeSelector Terms[].matchFields[].values

Description

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

Type

array

.spec.zookeeper.template.pod.affinity.nodeAffinity.require dDuringSchedulingIgnoredDuringExecution.nodeSelector Terms[].matchFields[].values[]

Type

string

.spec.zookeeper.template.pod.affinity.podAffinity

Description

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

Type

object

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, it may choose a node that violates one or more the expressions. The node that is most

Property	Type	Description
		<p>preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringSchedulingIgnoredDuringExecution affinity expressions, etc.) we compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has a label which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.</p>
<p><code>requiredDuringSchedulingIgnoredDuringExecution</code></p>	<p><code>array</code></p>	<p>If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to pod label update), the system may or may not try to eventually evict the pod from its node. When</p>

Property	Type	Description
		there are multiple elements, the lists of nodes corresponding each podAffinityTerm intersected, i.e. all terms must be satisfied.

.spec.zookeeper.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

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

Type

array

.spec.zookeeper.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

Description

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

Type

object

Required

podAffinityTerm

weight

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

.spec.zookeeper.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm

Description

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

Type

object

Required

topologyKey

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

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

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

`.spec.zookeeper.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector`

Description

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

Type

object

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

.spec.zookeeper.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions

Description

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

Type

array

.spec.zookeeper.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]

Description

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

Type

object

Required

key

operator

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

.spec.zookeeper.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

Description

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

Type

array

.spec.zookeeper.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]

Type

string

.spec.zookeeper.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels

Description

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

Type

object

.spec.zookeeper.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys

Description

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

Type

`array`

.spec.zookeeper.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]

Type

`string`

.spec.zookeeper.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys

Description

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

Type

`array`

.spec.zookeeper.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]

Type

`string`

.spec.zookeeper.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector

Description

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

Type

object

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

.spec.zookeeper.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions

Description

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

Type

array

`.spec.zookeeper.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]`

Description

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

Type

object

Required

key

operator

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

`.spec.zookeeper.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityT`

erm.namespaceSelector.matchExpressions[].values

Description

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

Type

array

.spec.zookeeper.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerms.namespaceSelector.matchExpressions[].values[]

Type

string

.spec.zookeeper.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerms.namespaceSelector.matchLabels

Description

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

Type

object

.spec.zookeeper.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerms.namespaces

Description

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

Type

array

.spec.zookeeper.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

string

.spec.zookeeper.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

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

Type

array

.spec.zookeeper.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

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

Type

object

Required

`topologyKey`

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

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

Property	Type	Description
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the <code>labelSelector</code> in the specified namespaces, where co-located is defined as running on a node whose value of the label with key <code>topologyKey</code> matches that of any node on which any of the selected pods is running. Empty <code>topologyKey</code> is not allowed.

`.spec.zookeeper.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector`

Description

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

Type

`object`

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

`.spec.zookeeper.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

Description

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

Type

array

`.spec.zookeeper.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]`

Description

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

Type

object

Required

key

operator

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

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

.spec.zookeeper.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values

Description

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

Type

array

.spec.zookeeper.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]

Type

string

.spec.zookeeper.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels

Description

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

Type

object

.spec.zookeeper.template.pod.affinity.podAffinity.required DuringSchedulingIgnoredDuringExecution[].matchLabelKeys

Description

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

Type

array

.spec.zookeeper.template.pod.affinity.podAffinity.required DuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]

Type

string

.spec.zookeeper.template.pod.affinity.podAffinity.required DuringSchedulingIgnoredDuringExecution[].mismatchLab

elKeys

Description

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

Type

array

`.spec.zookeeper.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]`

Type

string

`.spec.zookeeper.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector`

Description

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

Type

object

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

`.spec.zookeeper.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions`

Description

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

Type

`array`

`.spec.zookeeper.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]`

Description

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

Type

`object`

Required

key operator

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

.spec.zookeeper.template.pod.affinity.podAffinity.required DuringSchedulingIgnoredDuringExecution[].namespaceS elector.matchExpressions[].values

Description

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

Type

array

.spec.zookeeper.template.pod.affinity.podAffinity.required DuringSchedulingIgnoredDuringExecution[].namespaceS

`elector.matchExpressions[].values[]`

Type

string

`.spec.zookeeper.template.pod.affinity.podAffinity.required DuringSchedulingIgnoredDuringExecution[].namespaceS elector.matchLabels`

Description

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

Type

object

`.spec.zookeeper.template.pod.affinity.podAffinity.required DuringSchedulingIgnoredDuringExecution[].namespaces`

Description

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

Type

array

`.spec.zookeeper.template.pod.affinity.podAffinity.required DuringSchedulingIgnoredDuringExecution[].namespaces[]`

Type

string

.spec.zookeeper.template.pod.affinity.podAntiAffinity

Description

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

Type

object

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

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

`.spec.zookeeper.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution`

Description

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

Type

array

`.spec.zookeeper.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[]`

Description

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

Type

object

Required

podAffinityTerm

weight

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

.spec.zookeeper.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm

Description

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

Type

object

Required

topologyKey

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

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

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

`.spec.zookeeper.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector`

Description

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

Type

object

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

.spec.zookeeper.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions

Description

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

Type

array

.spec.zookeeper.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]

Description

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

Type

object

Required

key

operator

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

.spec.zookeeper.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

Description

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

Type

array

.spec.zookeeper.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]

Type

string

.spec.zookeeper.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels

Description

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

Type

object

.spec.zookeeper.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys

Description

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

Type

`array`

.spec.zookeeper.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]

Type

`string`

.spec.zookeeper.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys

Description

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

Type

`array`

.spec.zookeeper.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]

Type

`string`

`.spec.zookeeper.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector`

Description

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

Type

object

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

`.spec.zookeeper.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions`

Description

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

Type

array

`.spec.zookeeper.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]`

Description

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

Type

object

Required

key

operator

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

`.spec.zookeeper.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffi`

`nityTerm.namespaceSelector.matchExpressions[].values`

Description

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

Type

array

`.spec.zookeeper.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]`

Type

string

`.spec.zookeeper.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels`

Description

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

Type

object

`.spec.zookeeper.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces`

Description

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

Type

array

.spec.zookeeper.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

string

.spec.zookeeper.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

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

Type

array

.spec.zookeeper.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

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

Type

object

Required

`topologyKey`

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

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

Property	Type	Description
<code>topologyKey</code>	<code>string</code>	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

`.spec.zookeeper.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector`

Description

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

Type

`object`

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

`.spec.zookeeper.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions`

Description

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

Type

array

`.spec.zookeeper.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]`

Description

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

Type

object

Required

key

operator

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

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

.spec.zookeeper.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values

Description

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

Type

array

.spec.zookeeper.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]

Type

string

.spec.zookeeper.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels

Description

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

Type

object

.spec.zookeeper.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys

Description

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

Type

array

.spec.zookeeper.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]

Type

string

.spec.zookeeper.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatch

LabelKeys

Description

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

Type

array

`.spec.zookeeper.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]`

Type

string

`.spec.zookeeper.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector`

Description

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

Type

object

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

`.spec.zookeeper.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions`

Description

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

Type

`array`

`.spec.zookeeper.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]`

Description

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

Type

`object`

Required

key

operator

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

.spec.zookeeper.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

Description

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

Type

array

.spec.zookeeper.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespace

ceSelector.matchExpressions[].values[]

Type

string

.spec.zookeeper.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels

Description

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

Type

object

.spec.zookeeper.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces

Description

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

Type

array

.spec.zookeeper.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]

Type

string

.spec.zookeeper.template.pod.securityContext

Description

Security context for pods such as permissions and privilege escalation.

Type

object

Property	Type	Description
<code>appArmorProfile</code>	object	appArmorProfile is the AppArmor options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
<p><code>fsGroup</code></p>	<p><code>integer</code></p>	<p>A special supplemental group that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod:</p> <ol style="list-style-type: none"> 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---- <p>If unset, the Kubelet will not modify the ownership and permissions of any volume. Note that this field cannot be set when <code>spec.os.name</code> is windows.</p>
<p><code>fsGroupChangePolicy</code></p>	<p><code>string</code></p>	<p><code>fsGroupChangePolicy</code> defines behavior of changing ownership and permission of the volume before being exposed inside Pod. This field will only apply to volume types which support fsGroup based ownership(and permissions). It will have no effect on ephemeral volume types such as: secret, configmaps and emptydir. Valid values are "OnRootMismatch" and "Always". If not specified, "Always" is used. Note that this field cannot be set when <code>spec.os.name</code> is windows.</p>

Property	Type	Description
<code>runAsGroup</code>	<code>integer</code>	<p>The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.</p>
<code>runAsNonRoot</code>	<code>boolean</code>	<p>Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.</p>
<code>runAsUser</code>	<code>integer</code>	<p>The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.</p>

Property	Type	Description
<code>seLinuxChangePolicy</code>	<code>string</code>	<p><code>seLinuxChangePolicy</code> defines how the container's SELinux label is applied to all volumes used by the Pod. It has no effect on nodes that do not support SELinux or to volumes does not support SELinux. Valid values are "MountOption" and "Recursive".</p> <p>"Recursive" means relabeling of all files on all Pod volumes by the container runtime. This may be slow for large volumes, but allows mixing privileged and unprivileged Pods sharing the same volume on the same node.</p> <p>"MountOption" mounts all eligible Pod volumes with <code>-o context</code> mount option. This requires all Pods that share the same volume to use the same SELinux label. It is not possible to share the same volume among privileged and unprivileged Pods. Eligible volumes are in-tree FibreChannel and iSCSI volumes, and all CSI volumes whose CSI driver announces SELinux support by setting <code>spec.seLinuxMount: true</code> in their CSIDriver instance. Other volumes are always re-labelled recursively.</p> <p>"MountOption" value is allowed only when SELinuxMount feature gate is enabled.</p> <p>If not specified and SELinuxMount feature gate is enabled, "MountOption" is used. If not specified and SELinuxMount feature gate is disabled, "MountOption" is used for ReadWriteOncePod volumes and "Recursive" for all other volumes.</p>

Property	Type	Description
		<p>This field affects only Pods that have SELinux label set, either in PodSecurityContext or in SecurityContext of all containers.</p> <p>All Pods that use the same volume should use the same seLinuxChangePolicy, otherwise some pods can get stuck in ContainerCreating state. Note that this field cannot be set when spec.os.name is windows.</p>
seLinuxOptions	object	<p>The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.</p>
seccompProfile	object	<p>The seccomp options to use by the containers in this pod. Note that this field cannot be set when spec.os.name is windows.</p>
supplementalGroups	array	<p>A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if</p>

Property	Type	Description
		specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.
<code>supplementalGroupsPolicy</code>	<code>string</code>	Defines how supplemental groups of the first container processes are calculated. Valid values are "Merge" and "Strict". If not specified, "Merge" is used. (Alpha) Using the field requires the SupplementalGroupsPolicy feature gate to be enabled and the container runtime must implement support for this feature. Note that this field cannot be set when spec.os.name is windows.
<code>sysctls</code>	<code>array</code>	Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.

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

`.spec.zookeeper.template.pod.securityContext.appArmorProfile`

Description

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

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if <code>type</code> is "Localhost".

Property	Type	Description
<code>type</code>	<code>string</code>	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.

`.spec.zookeeper.template.pod.securityContext.seLinuxOptions`

Description

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

Type

`object`

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

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

`.spec.zookeeper.template.pod.securityContext.seccompProfile`

Description

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

Type

`object`

Required

`type`

Property	Type	Description
<code>localhostProfile</code>	<code>string</code>	<code>localhostProfile</code> indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if <code>type</code> is "Localhost". Must NOT be set for any other type.

Property	Type	Description
type	string	<p>type indicates which kind of seccomp profile will be applied. Valid options are:</p> <p>Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied.</p>

.spec.zookeeper.template.pod.securityContext.supplementalGroups

Description

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

Type

array

.spec.zookeeper.template.pod.securityContext.supplementalGroups[]

Type

integer

.spec.zookeeper.template.pod.securityContext.sysctls

Description

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

Type

array

`.spec.zookeeper.template.pod.securityContext.sysctls[]`

Description

Sysctl defines a kernel parameter to be set

Type

object

Required

name

value

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

`.spec.zookeeper.template.pod.securityContext.windowsOptions`

Description

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

Type

object

Property	Type	Description
<code>gmsaCredentialSpec</code>	<code>string</code>	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
<code>gmsaCredentialSpecName</code>	<code>string</code>	GMSACredentialSpecName is the name of the GMSA credential spec to use.
<code>hostProcess</code>	<code>boolean</code>	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.

Property	Type	Description
<code>runAsUserName</code>	<code>string</code>	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

`.spec.zookeeper.template.pod.tolerations`

Description

Tolerations for pods to schedule onto nodes with taints.

Type

`array`

`.spec.zookeeper.template.pod.tolerations[]`

Description

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

Type

`object`

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

.status

Description

RdsKafkaStatus defines the observed state of RdsKafka

Type

object

Required

lastUpdateTime

phase

Property	Type	Description
clusterId	string	Kafka ClusterId
conditions	array	
lastUpdateTime	string	
migration	string	Migration phase
nodes	array	
operatorVersion	string	
phase	string	INSERT ADDITIONAL STATUS FIELD - define observed state of cluster Important: Run "make" to regenerate code after modifying this file
pvcMatchLabels	object	
reason	string	
servicesName	object	

Property	Type	Description
<code>strimziKafkaName</code>	<code>string</code>	
<code>upgradeStatus</code>	<code>object</code>	UpgradeStatus indicates the status of the bundle upgrade.

`.status.conditions`

Type

`array`

`.status.conditions[]`

Type

`object`

Property	Type	Description
<code>lastTransitionTime</code>	<code>string</code>	Last time the condition of a type changed from one status to another. The required format is 'yyyy-MM-ddTHH:mm:ssZ', in the UTC time zone.
<code>message</code>	<code>string</code>	Human-readable message indicating details about the condition's last transition.
<code>reason</code>	<code>string</code>	The reason for the condition's last transition (a single word in CamelCase).

Property	Type	Description
<code>status</code>	<code>string</code>	The status of the condition, either True, False or Unknown.
<code>type</code>	<code>string</code>	The unique identifier of a condition, used to distinguish between other conditions in the resource.

.status.nodes

Type

`array`

.status.nodes[]

Type

`object`

Property	Type	Description
<code>ipAddress</code>	<code>string</code>	
<code>nodeName</code>	<code>string</code>	
<code>pod</code>	<code>string</code>	

.status.pvcMatchLabels

Type

`object`

.status.servicesName

Type

object

.status.upgradeStatus

Description

UpgradeStatus indicates the status of the bundle upgrade.

Type

object

Property	Type	Description
crVersion	string	CRVersion is the version of the CR

API Endpoints

The following API endpoints are available:

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

- `/apis/middleware.alauda.io/v1/namespaces/{namespace}/rdskafkas/{name}/status`
 - `GET` : read status of the specified RdsKafka
 - `PATCH` : partially update status of the specified RdsKafka
 - `PUT` : replace status of the specified RdsKafka

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

HTTP method

`DELETE`

Description

delete collection of RdsKafka

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind RdsKafka

HTTP responses

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

HTTP method

POST

Description

create a new RdsKafka

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>RdsKafka</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>RdsKafka</code> schema
201 - Created	<code>RdsKafka</code> schema
202 - Accepted	<code>RdsKafka</code> schema
401 - Unauthorized	Empty

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

HTTP method

DELETE

Description

delete the specified RdsKafka

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

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

HTTP method

GET

Description

read the specified RdsKafka

HTTP responses

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

HTTP method

PATCH

Description

partially update the specified RdsKafka

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields.

Parameter	Type	Description
		This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

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

HTTP method

PUT

Description

replace the specified RdsKafka

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object,

Parameter	Type	Description
		and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	RdsKafka schema	application/json formatted

HTTP responses

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

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

HTTP method

GET

Description

read status of the specified RdsKafka

HTTP responses

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

HTTP method

PATCH

Description

partially update status of the specified RdsKafka

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

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

HTTP method

PUT

Description

replace status of the specified RdsKafka

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	<p><code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are:</p> <ul style="list-style-type: none"> - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+. - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>RdsKafka</code> schema	<code>application/json</code> formatted

HTTP responses

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

RdsTopic

Description

RdsTopic is the Schema for the rdstopics API

Type

object

Specification

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

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

.spec

Description

RdsTopicSpec defines the desired state of RdsTopic

Type

object

Property	Type	Description
config	object	Config is a map of topic configuration properties

Property	Type	Description
<code>partitions</code>	<code>integer</code>	Partitions is the number of partitions for the topic
<code>replicas</code>	<code>integer</code>	Replicas is the replication factor, number of replicas per partition
<code>topicName</code>	<code>string</code>	TopicName is the name of the topic

.spec.config

Description

Config is a map of topic configuration properties

Type

`object`

.status

Description

RdsTopicStatus defines the observed state of RdsTopic

Type

`object`

Required

`lastUpdateTime` `phase`

Property	Type	Description
<code>lastUpdateTime</code>	<code>string</code>	

Property	Type	Description
phase	string	INSERT ADDITIONAL STATUS FIELD - define observed state of cluster Important: Run "make" to regenerate code after modifying this file
reason	string	
strimziKafkaTopicName	string	

API Endpoints

The following API endpoints are available:

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

/apis/middleware.alauda.io/v1/namespaces/{namespace}/rds/topics

HTTP method

DELETE

Description

delete collection of RdsTopic

HTTP responses

HTTP code	Response body
200 - OK	Status schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind RdsTopic

HTTP responses

HTTP code	Response body
200 - OK	RdsTopicList schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new RdsTopic

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>RdsTopic</code> schema	<code>application/json</code> formatted

HTTP responses

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

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

/apis/middleware.alauda.io/v1/namespaces/{namespace}/rds/topics/{name}

HTTP method

DELETE

Description

delete the specified RdsTopic

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

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

HTTP method

GET

Description

read the specified RdsTopic

HTTP responses

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

HTTP method

PATCH

Description

partially update the specified RdsTopic

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server

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

HTTP responses

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

HTTP method

PUT

Description

replace the specified RdsTopic

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the

Parameter	Type	Description
		request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	RdsTopic schema	application/json formatted

HTTP responses

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

/apis/middleware.alauda.io/v1/namespaces/{namespace}/rds/topics/{name}/status

HTTP method

GET

Description

read status of the specified RdsTopic

HTTP responses

HTTP code	Response body
200 - OK	RdsTopic schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified RdsTopic

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

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

HTTP method

PUT

Description

replace status of the specified RdsTopic

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	RdsTopic schema	application/json formatted

HTTP responses

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

RdsKafkaUser

Description

RdsKafkaUser is the Schema for the rdskafkausers API

Type

object

Specification

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

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

.spec

Description

RdsKafkaUserSpec defines the desired state of RdsKafkaUser

Type

object

Required

authentication

authorization

Property	Type	Description
authentication	object	Authentication configuration for the Kafka user
authorization	object	Authorization configuration controlling Kafka permissions

.spec.authentication

Description

Authentication configuration for the Kafka user

Type

object

Required

type

Property	Type	Description
password	object	Password secret reference
type	string	

.spec.authentication.password

Description

Password secret reference

Type

object

Required

`valueFrom`

Property	Type	Description
<code>valueFrom</code>	<code>object</code>	Secret key ref for the password

`.spec.authentication.password.valueFrom`

Description

Secret key ref for the password

Type

`object`

Required

`secretKeyRef`

Property	Type	Description
<code>secretKeyRef</code>	<code>object</code>	Secret key selector

`.spec.authentication.password.valueFrom.secretKeyRef`

Description

Secret key selector

Type

`object`

Required

`key`

Property	Type	Description
key	string	The key of the secret to select from. Must be a valid secret key.
name	string	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names [↗] TODO: Add other useful fields. apiVersion, kind, uid?
optional	boolean	Specify whether the Secret or its key must be defined

.spec.authorization

Description

Authorization configuration controlling Kafka permissions

Type

object

Required

acls

type

Property	Type	Description
acls	array	Access control lists
type	string	

.spec.authorization.acls

Description

Access control lists

Type

array

.spec.authorization.acls[]

Type

object

Required

host

operation

resource

Property	Type	Description
host	string	Host is the host pattern that applies to the ACL
operation	string	Operation is one of the allowed Kafka operation types
resource	object	Resource defines the type and name of the resource

.spec.authorization.acls[].resource

Description

Resource defines the type and name of the resource

Type

object

Required

name

patternType

type

Property	Type	Description
<code>name</code>	<code>string</code>	Name is the resource name
<code>patternType</code>	<code>string</code>	PatternType is how the name should be interpreted
<code>type</code>	<code>string</code>	Type is the type of resource

.status

Description

RdsKafkaUserStatus defines the observed state of RdsKafkaUser

Type

`object`

Required

`lastUpdateTime` `phase`

Property	Type	Description
<code>lastUpdateTime</code>	<code>string</code>	
<code>phase</code>	<code>string</code>	INSERT ADDITIONAL STATUS FIELD - define observed state of cluster Important: Run "make" to regenerate code after modifying this file
<code>reason</code>	<code>string</code>	
<code>secret</code>	<code>string</code>	
<code>strimziKafkaUserName</code>	<code>string</code>	

Property	Type	Description
username	string	

API Endpoints

The following API endpoints are available:

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

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

HTTP method

DELETE

Description

delete collection of RdsKafkaUser

HTTP responses

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

HTTP method

GET

Description

list objects of kind RdsKafkaUser

HTTP responses

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

HTTP method

POST

Description

create a new RdsKafkaUser

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing

Parameter	Type	Description
		unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	RdsKafkaUser schema	application/json formatted

HTTP responses

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

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

HTTP method

DELETE

Description

delete the specified RdsKafkaUser

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
202 - Accepted	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified RdsKafkaUser

HTTP responses

HTTP code	Response body
200 - OK	RdsKafkaUser schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified RdsKafkaUser

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

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

HTTP method

PUT

Description

replace the specified RdsKafkaUser

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>RdsKafkaUser</code> schema	<code>application/json</code> formatted

HTTP responses

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

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

HTTP method

`GET`

Description

read status of the specified RdsKafkaUser

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified RdsKafkaUser

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun

Parameter	Type	Description
		directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

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

HTTP method

`PUT`

Description

replace status of the specified RdsKafkaUser

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>RdsKafkaUser</code> schema	<code>application/json</code> formatted

HTTP responses

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

HTTP code	Response body
401 - Unauthorized	Empty

RdsMirrorMaker2

Description

RdsMirrorMaker2 is the Schema for the rdsmirrormaker2 API

Type

object

Specification

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

Property	Type	Description
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	RdsMirrorMaker2Spec defines the desired state of RdsMirrorMaker2
status	object	RdsKafkaStatus defines the observed state of RdsKafka

.spec

Description

RdsMirrorMaker2Spec defines the desired state of RdsMirrorMaker2

Type

object

Property	Type	Description
clusters	array	

Property	Type	Description
<code>jvmOptions</code>	<code>object</code>	JVM options such as Xms and Xmx
<code>logging</code>	<code>object</code>	
<code>mirrors</code>	<code>array</code>	
<code>replicas</code>	<code>integer</code>	Number of replicas for MirrorMaker2 replica
<code>resources</code>	<code>object</code>	Resource requests/limits for pods

`.spec.clusters`

Type

`array`

`.spec.clusters[]`

Type

`object`

Required

`bootstrapServers`

Property	Type	Description
<code>authentication</code>	<code>object</code>	
<code>bootstrapServers</code>	<code>string</code>	
<code>tls</code>	<code>object</code>	

`.spec.clusters[].authentication`

Type

object

Required

type

Property	Type	Description
<code>certificateAndKey</code>	object	
<code>passwordSecret</code>	object	
<code>type</code>	string	
<code>username</code>	string	

`.spec.clusters[].authentication.certificateAndKey`

Type

object

Required

certificate

key

secretName

Property	Type	Description
<code>certificate</code>	string	
<code>key</code>	string	
<code>secretName</code>	string	

`.spec.clusters[].authentication.passwordSecret`

Type

object

Required

password

secretName

Property	Type	Description
password	string	
secretName	string	

`.spec.clusters[].tls`

Type

object

Required

trustedCertificates

Property	Type	Description
trustedCertificates	array	

`.spec.clusters[].tls.trustedCertificates`

Type

array

`.spec.clusters[].tls.trustedCertificates[]`

Type

object

Required

pattern

secretName

Property	Type	Description
<code>pattern</code>	<code>string</code>	
<code>secretName</code>	<code>string</code>	

`.spec.jvmOptions`

Description

JVM options such as Xms and Xmx

Type

`object`

Property	Type	Description
<code>-Xms</code>	<code>string</code>	The Xms field sets the JVM min heap size parameter
<code>-Xmx</code>	<code>string</code>	The Xmx field sets the JVM max heap size parameter

`.spec.logging`

Type

`object`

Required

`loggers`

`type`

Property	Type	Description
<code>loggers</code>	<code>object</code>	Loggers is a map from logger name to logger level.

Property	Type	Description
type	string	Logging type , it must have the value inline

.spec.logging.loggers

Description

Loggers is a map from logger name to logger level.

Type

object

.spec.mirrors

Type

array

.spec.mirrors[]

Type

object

Property	Type	Description
checkpointConnector	object	
groupsPattern	string	
sourceConnector	object	
topicsPattern	string	

.spec.mirrors[].checkpointConnector

Type

object

Property	Type	Description
config	object	
tasksMax	integer	

.spec.mirrors[].checkpointConnector.config**Type**

object

.spec.mirrors[].sourceConnector**Type**

object

Property	Type	Description
config	object	
tasksMax	integer	

.spec.mirrors[].sourceConnector.config**Type**

object

.spec.resources**Description**

Resource requests/limits for pods

Type

object

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

`.spec.resources.claims`

Description

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

Type

array

.spec.resources.claims[]

Description

ResourceClaim references one entry in PodSpec.ResourceClaims.

Type

object

Required

name

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

.spec.resources.limits

Description

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

Type

object

.spec.resources.requests

Description

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

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

Type

object

.status

Description

RdsKafkaStatus defines the observed state of RdsKafka

Type

object

Required

lastUpdateTime

phase

Property	Type	Description
conditions	array	
lastUpdateTime	string	
phase	string	INSERT ADDITIONAL STATUS FIELD - define observed state of cluster Important: Run "make" to regenerate code after modifying this file
reason	string	

.status.conditions

Type

array

.status.conditions[]

Type

object

Property	Type	Description
<code>lastTransitionTime</code>	<code>string</code>	Last time the condition of a type changed from one status to another. The required format is 'yyyy-MM-ddTHH:mm:ssZ', in the UTC time zone.
<code>message</code>	<code>string</code>	Human-readable message indicating details about the condition's last transition.
<code>reason</code>	<code>string</code>	The reason for the condition's last transition (a single word in CamelCase).
<code>status</code>	<code>string</code>	The status of the condition, either True, False or Unknown.
<code>type</code>	<code>string</code>	The unique identifier of a condition, used to distinguish between other conditions in the resource.

API Endpoints

The following API endpoints are available:

- `/apis/middleware.alauda.io/v1/namespaces/{namespace}/rdsmirrormaker2s`
 - **DELETE** : delete collection of RdsMirrorMaker2
 - **GET** : list objects of kind RdsMirrorMaker2
 - **POST** : create a new RdsMirrorMaker2
- `/apis/middleware.alauda.io/v1/namespaces/{namespace}/rdsmirrormaker2s/{name}`
 - **DELETE** : delete the specified RdsMirrorMaker2
 - **GET** : read the specified RdsMirrorMaker2
 - **PATCH** : partially update the specified RdsMirrorMaker2
 - **PUT** : replace the specified RdsMirrorMaker2
- `/apis/middleware.alauda.io/v1/namespaces/{namespace}/rdsmirrormaker2s/{name}/status`
 - **GET** : read status of the specified RdsMirrorMaker2
 - **PATCH** : partially update status of the specified RdsMirrorMaker2
 - **PUT** : replace status of the specified RdsMirrorMaker2

`/apis/middleware.alauda.io/v1/namespaces/{namespace}/rdsmirrormaker2s`

HTTP method

DELETE

Description

delete collection of RdsMirrorMaker2

HTTP responses

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

HTTP method

GET

Description

list objects of kind RdsMirrorMaker2

HTTP responses

HTTP code	Response body
200 - OK	<code>RdsMirrorMaker2List</code> schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new RdsMirrorMaker2

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last

Parameter	Type	Description
		<p>duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.</p>

Body parameters

Parameter	Type	Description
body	RdsMirrorMaker2 schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	RdsMirrorMaker2 schema
201 - Created	RdsMirrorMaker2 schema
202 - Accepted	RdsMirrorMaker2 schema
401 - Unauthorized	Empty

/apis/middleware.alauda.io/v1/namespaces/{namespace}/rdsmirrorMaker2s/{name}

HTTP method

DELETE

Description

delete the specified RdsMirrorMaker2

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

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

HTTP method

`GET`

Description

read the specified RdsMirrorMaker2

HTTP responses

HTTP code	Response body
200 - OK	<code>RdsMirrorMaker2</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified RdsMirrorMaker2

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>RdsMirrorMaker2</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified RdsMirrorMaker2

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>RdsMirrorMaker2</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>RdsMirrorMaker2</code> schema
201 - Created	<code>RdsMirrorMaker2</code> schema
401 - Unauthorized	Empty

/apis/middleware.alauda.io/v1/namespaces/{namespace}/rdsmirrorMaker2s/{name}/status

HTTP method

GET

Description

read status of the specified RdsMirrorMaker2

HTTP responses

HTTP code	Response body
200 - OK	<code>RdsMirrorMaker2</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified RdsMirrorMaker2

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further

Parameter	Type	Description
		processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<p><code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.</p>

HTTP responses

HTTP code	Response body
200 - OK	<code>RdsMirrorMaker2</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified RdsMirrorMaker2

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>RdsMirrorMaker2</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>RdsMirrorMaker2</code> schema
201 - Created	<code>RdsMirrorMaker2</code> schema

HTTP code	Response body
401 - Unauthorized	Empty