Triggers

Trigger [triggers.tekton.dev/v1b
TriggerTemplate [triggers.tekto
EventLister

TriggerBinding [triggers.tekton
Interceptor [triggers.tekton.dev
ClusterTrig

ClusterInterceptor [triggers.tek
ScheduledTrigger [tekton.alaudadevops.io/v1

Trigger [triggers.tekton.dev/v1beta1]

Description

Trigger defines a mapping of an input event to parameters. This is used to extract information from events to be passed to TriggerTemplates within a Trigger.

Type

`object`

Specification

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

Property	Type	Description
		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`	Spec holds the desired state of the Trigger

.spec

Description

Spec holds the desired state of the Trigger

Type

`object`

Required

`bindings` `template`

Property	Type	Description
`bindings`	`array`	
`interceptors`	`array`	
`name`	`string`	

Property	Type	Description
`serviceAccountName`	`string`	ServiceAccountName optionally associates credentials with each trigger; Unlike EventListeners, this should be scoped to the same namespace as the Trigger itself
`template`	`object`	

.spec.bindings

Type

`array`

.spec.bindings[]

Type

`object`

Property	Type	Description
`apiversion`	`string`	APIVersion of the binding ref
`kind`	`string`	Kind can only be provided if Ref is also provided. Defaults to TriggerBinding
`name`	`string`	Name is the name of the binding param Mutually exclusive with Ref

Property	Type	Description
`ref`	`string`	Ref is a reference to a TriggerBinding kind. Mutually exclusive with Name
`value`	`string`	Value is the value of the binding param. Can contain JSONPath Has to be pointer since "" is a valid value Required if Name is also specified.

.spec.interceptors

Type

`array`

.spec.interceptors[]

Description

TriggerInterceptor provides a hook to intercept and pre-process events

Type

`object`

Required

`ref`

Property	Type	Description
`bitbucket`	`object`	BitbucketInterceptor provides a webhook to intercept and pre-process events

Property	Type	Description
`cel`	`object`	CELInterceptor provides a webhook to intercept and pre-process events
`github`	`object`	Deprecated old fields below
`gitlab`	`object`	GitLabInterceptor provides a webhook to intercept and pre-process events
`name`	`string`	Optional name to identify the current interceptor configuration
`params`	`array`	Params are the params to send to the interceptor
`ref`	`object`	Ref refers to the Interceptor to use
`webhook`	`object`	WebhookInterceptor refers to an old style webhook interceptor service

.spec.interceptors[].bitbucket

Description

BitbucketInterceptor provides a webhook to intercept and pre-process events

Type

`object`

Property	Type	Description
`eventTypes`	`array`	
`secretRef`	`object`	SecretRef contains the information required to reference a single secret string This is needed because the other secretRef types are not cross-namespace and do not actually contain the "SecretName" field, which allows us to access a single secret value.

.spec.interceptors[].bitbucket.eventTypes

Type

`array`

.spec.interceptors[].bitbucket.eventTypes[]

Type

`string`

.spec.interceptors[].bitbucket.secretRef

Description

SecretRef contains the information required to reference a single secret string This is needed because the other secretRef types are not cross-namespace and do not actually contain the "SecretName" field, which allows us to access a single secret value.

Type

`object`

Property	Type	Description
`secretKey`	`string`	

Property	Type	Description
`secretName`	`string`	

.spec.interceptors[].cel

Description

CELInterceptor provides a webhook to intercept and pre-process events

Type

`object`

Property	Type	Description
`filter`	`string`	
`overlays`	`array`	

.spec.interceptors[].cel.overlays

Type

`array`

.spec.interceptors[].cel.overlays[]

Description

CELOverlay provides a way to modify the request body using DeprecatedCEL expressions

Type

`object`

Property	Type	Description
`expression`	`string`	

Property	Type	Description
`key`	`string`	

.spec.interceptors[].github

Description

Deprecated old fields below

Type

`object`

Property	Type	Description
`eventTypes`	`array`	
`secretRef`	`object`	SecretRef contains the information required to reference a single secret string This is needed because the other secretRef types are not cross-namespace and do not actually contain the "SecretName" field, which allows us to access a single secret value.

.spec.interceptors[].github.eventTypes

Type

`array`

.spec.interceptors[].github.eventTypes[]

Type

`string`

.spec.interceptors[].github.secretRef

Description

SecretRef contains the information required to reference a single secret string This is needed because the other secretRef types are not cross-namespace and do not actually contain the "SecretName" field, which allows us to access a single secret value.

Type

`object`

Property	Type	Description
`secretKey`	`string`	
`secretName`	`string`	

.spec.interceptors[].gitlab

Description

GitLabInterceptor provides a webhook to intercept and pre-process events

Type

`object`

Property	Type	Description
`eventTypes`	`array`	
`secretRef`	`object`	SecretRef contains the information required to reference a single secret string This is needed because the other secretRef types are not cross-namespace and do not actually contain the "SecretName" field, which allows us to access a single secret value.

.spec.interceptors[].gitlab.eventTypes

Type

`array`

.spec.interceptors[].gitlab.eventTypes[]

Type

`string`

.spec.interceptors[].gitlab.secretRef

Description

SecretRef contains the information required to reference a single secret string This is needed because the other secretRef types are not cross-namespace and do not actually contain the "SecretName" field, which allows us to access a single secret value.

Type

`object`

Property	Type	Description
`secretKey`	`string`	
`secretName`	`string`	

.spec.interceptors[].params

Description

Params are the params to send to the interceptor

Type

`array`

.spec.interceptors[].params[]

Description

InterceptorParams defines a key-value pair that can be passed on an interceptor

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	
`value`		

.spec.interceptors[].ref

Description

Ref refers to the Interceptor to use

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent
`kind`	`string`	InterceptorKind indicates the kind of the Interceptor, namespaced or cluster scoped.
`name`	`string`	Name of the referent; More info: http://kubernetes.io/docs/user-guide/identifiers#names ↗

.spec.interceptors[].webhook

Description

WebhookInterceptor refers to an old style webhook interceptor service

Type

`object`

Property	Type	Description
`header`	`array`	Header is a group of key-value pairs that can be appended to the interceptor request headers. This allows the interceptor to make decisions specific to an EventListenerTrigger.
`objectRef`	`object`	ObjectRef is a reference to an object that will resolve to a cluster DNS name to use as the EventInterceptor. Either objectRef or url can be specified
`url`	`string`	

.spec.interceptors[].webhook.header

Description

Header is a group of key-value pairs that can be appended to the interceptor request headers. This allows the interceptor to make decisions specific to an EventListenerTrigger.

Type

`array`

.spec.interceptors[].webhook.header[]

Description

Param declares an ParamValues to use for the parameter called name.

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	
`value`		

.spec.interceptors[].webhook.objectRef

Description

ObjectRef is a reference to an object that will resolve to a cluster DNS name to use as the EventInterceptor. Either objectRef or url can be specified

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.

Property	Type	Description
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗
`resourceVersion`	`string`	Specific resourceVersion to which this reference is made, if any. More info: https://git.k8s.io/community/contributors/devel/sig-

Property	Type	Description
		architecture/api-conventions.md#concurrency-control-and-consistency ↗
`uid`	`string`	UID of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#uids ↗

.spec.template

Type

`object`

Property	Type	Description
`apiversion`	`string`	
`ref`	`string`	
`spec`	`object`	TriggerTemplateSpec holds the desired state of TriggerTemplate

.spec.template.spec

Description

TriggerTemplateSpec holds the desired state of TriggerTemplate

Type

`object`

Property	Type	Description
`params`	`array`	
`resourcetemplates`	`array`	

.spec.template.spec.params

Type

`array`

.spec.template.spec.params[]

Description

ParamSpec defines an arbitrary named input whose value can be supplied by a `Param`.

Type

`object`

Required

`name`

Property	Type	Description
`default`	`string`	Default is the value a parameter takes if no input value via a Param is supplied.
`description`	`string`	Description is a user-facing description of the parameter that may be used to populate a UI.
`name`	`string`	Name declares the name by which a parameter is referenced.

.spec.template.spec.resourcetemplates

Type

`array`

.spec.template.spec.resourcetemplates[]

Description

TriggerResourceTemplate describes a resource to create

Type

`object`

API Endpoints

The following API endpoints are available:

- `/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/triggers`
  - `DELETE`: delete collection of Trigger
  - `GET`: list objects of kind Trigger
  - `POST`: create a new Trigger
- `/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/triggers/{name}`
  - `DELETE`: delete the specified Trigger
  - `GET`: read the specified Trigger
  - `PATCH`: partially update the specified Trigger
  - `PUT`: replace the specified Trigger

/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/triggers

HTTP method

`DELETE`

Description

delete collection of Trigger

HTTP responses

HTTP code	Response body
200 - OK	`Status` schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

list objects of kind Trigger

HTTP responses

HTTP code	Response body
200 - OK	`TriggerList` schema
401 - Unauthorized	Empty

HTTP method

`POST`

Description

create a new Trigger

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

Parameter	Type	Description
		processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`Trigger` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`Trigger` schema
201 - Created	`Trigger` schema
202 - Accepted	`Trigger` schema
401 - Unauthorized	Empty

/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/triggers/{name}

HTTP method

`DELETE`

Description

delete the specified Trigger

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 Trigger

HTTP responses

HTTP code	Response body
200 - OK	`Trigger` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified Trigger

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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	`Trigger` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified Trigger

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

Parameter	Type	Description
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`Trigger` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`Trigger` schema
201 - Created	`Trigger` schema
401 - Unauthorized	Empty

TriggerTemplate [triggers.tekton.dev/v1beta1]

Description

TriggerTemplate takes parameters and uses them to create CRDs

Type

`object`

Specification

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

Property	Type	Description
		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`	Spec holds the desired state of the TriggerTemplate from the client
`status`	`object`	TriggerTemplateStatus describes the desired state of TriggerTemplate

.spec

Description

Spec holds the desired state of the TriggerTemplate from the client

Type

`object`

Property	Type	Description
`params`	`array`	
`resourcetemplates`	`array`	

.spec.params

Type

`array`

.spec.params[]

Description

ParamSpec defines an arbitrary named input whose value can be supplied by a `Param`.

Type

`object`

Required

`name`

Property	Type	Description
`default`	`string`	Default is the value a parameter takes if no input value via a Param is supplied.
`description`	`string`	Description is a user-facing description of the parameter that may be used to populate a UI.
`name`	`string`	Name declares the name by which a parameter is referenced.

.spec.resourcetemplates

Type

`array`

.spec.resourcetemplates[]

Description

TriggerResourceTemplate describes a resource to create

Type

`object`

.status

Description

TriggerTemplateStatus describes the desired state of TriggerTemplate

Type

`object`

API Endpoints

The following API endpoints are available:

- `/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/triggertemplates`
  - `DELETE`: delete collection of TriggerTemplate
  - `GET`: list objects of kind TriggerTemplate
  - `POST`: create a new TriggerTemplate
- `/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/triggertemplates/{name}`
  - `DELETE`: delete the specified TriggerTemplate
  - `GET`: read the specified TriggerTemplate
  - `PATCH`: partially update the specified TriggerTemplate
  - `PUT`: replace the specified TriggerTemplate
- `/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/triggertemplates/{name}/status`
  - `GET`: read status of the specified TriggerTemplate

- `PATCH`: partially update status of the specified TriggerTemplate
- `PUT`: replace status of the specified TriggerTemplate

/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/triggertemplates

HTTP method

`DELETE`

Description

delete collection of TriggerTemplate

HTTP responses

HTTP code	Response body
200 - OK	`Status` schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

list objects of kind TriggerTemplate

HTTP responses

HTTP code	Response body
200 - OK	`TriggerTemplateList` schema
401 - Unauthorized	Empty

HTTP method

`POST`

Description

create a new TriggerTemplate

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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`TriggerTemplate` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`TriggerTemplate` schema

HTTP code	Response body
201 - Created	`TriggerTemplate` schema
202 - Accepted	`TriggerTemplate` schema
401 - Unauthorized	Empty

/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/triggertemplates/{name}

HTTP method

`DELETE`

Description

delete the specified TriggerTemplate

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 TriggerTemplate

HTTP responses

HTTP code	Response body
200 - OK	`TriggerTemplate` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified TriggerTemplate

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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the

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.

HTTP responses

HTTP code	Response body
200 - OK	`TriggerTemplate` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified TriggerTemplate

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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The

Parameter	Type	Description
		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`	`TriggerTemplate` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`TriggerTemplate` schema
201 - Created	`TriggerTemplate` schema
401 - Unauthorized	Empty

/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/triggertemplates/{name}/status

HTTP method

`GET`

Description

read status of the specified TriggerTemplate

HTTP responses

HTTP code	Response body
200 - OK	`TriggerTemplate` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified TriggerTemplate

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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	`TriggerTemplate` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified TriggerTemplate

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

Parameter	Type	Description
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`TriggerTemplate` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`TriggerTemplate` schema
201 - Created	`TriggerTemplate` schema
401 - Unauthorized	Empty

EventListener [triggers.tekton.dev/v1beta1]

Description

EventListener exposes a service to accept HTTP event payloads.

Type

`object`

Specification

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

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

.spec

Description

Spec holds the desired state of the EventListener from the client

Type

`object`

Required

`triggers`

Property	Type	Description
`labelSelector`	`object`	A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.
`namespaceSelector`	`object`	NamespaceSelector is a selector for selecting either all namespaces or a list of namespaces.
`resources`	`object`	
`serviceAccountName`	`string`	
`triggers`	`array`	

.spec.labelSelector

Description

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

Type

`object`

Property	Type	Description
`matchExpressions`	`array`	matchExpressions is a list of label selector requirements. The requirements are ANDed.

Property	Type	Description
`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.labelSelector.matchExpressions

Description

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

Type

`array`

.spec.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
`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.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.labelSelector.matchExpressions[].values[]

Type

`string`

.spec.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.namespaceSelector

Description

NamespaceSelector is a selector for selecting either all namespaces or a list of namespaces.

Type

`object`

Property	Type	Description
`matchNames`	`array`	List of namespace names.

.spec.namespaceSelector.matchNames

Description

List of namespace names.

Type

`array`

.spec.namespaceSelector.matchNames[]

Type

`string`

.spec.resources

Type

`object`

Property	Type	Description
`customResource`	`object`	
`kubernetesResource`	`object`	

.spec.resources.customResource

Type

`object`

.spec.resources.kubernetesResource

Type

`object`

Property	Type	Description
`replicas`	`integer`	
`serviceType`	`string`	Service Type string describes ingress methods for a service
`spec`	`object`	WithPodSpec is the shell around the PodSpecable within WithPod.

.spec.resources.kubernetesResource.spec

Description

WithPodSpec is the shell around the PodSpecable within WithPod.

Type

`object`

Property	Type	Description
`template`	`object`	PodSpecable is implemented by types containing a PodTemplateSpec in the manner of ReplicaSet, Deployment, DaemonSet, StatefulSet.

.spec.resources.kubernetesResource.spec.template

Description

PodSpecable is implemented by types containing a PodTemplateSpec in the manner of ReplicaSet, Deployment, DaemonSet, StatefulSet.

Type

`object`

Property	Type	Description
`metadata`	`ObjectMeta` ↗	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata ↗
`spec`	`object`	Specification of the desired behavior of the pod. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status ↗

.spec.resources.kubernetesResource.spec.template.spec

Description

Specification of the desired behavior of the pod. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status

Type

`object`

Required

`containers`

Property	Type	Description
`activeDeadlineSeconds`	`integer`	Optional duration in seconds the pod may StartTime before the system will actively tr containers. Value must be a positive intege
`affinity`	`object`	If specified, the pod's scheduling constrain
`automountServiceAccountToken`	`boolean`	AutomountServiceAccountToken indicates be automatically mounted.
`containers`	`array`	List of containers belonging to the pod. Co removed. There must be at least one conta
`dnsConfig`	`object`	Specifies the DNS parameters of a pod. Pa to the generated DNS configuration based
`dnsPolicy`	`string`	Set DNS policy for the pod. Defaults to "Cl 'ClusterFirstWithHostNet', 'ClusterFirst', 'De in DNSConfig will be merged with the polic

Property	Type	Description
		DNS options set along with hostNetwork, y to 'ClusterFirstWithHostNet'.
`enableServiceLinks`	`boolean`	EnableServiceLinks indicates whether info injected into pod's environment variables, r Optional: Defaults to true.
`ephemeralContainers`	`array`	List of ephemeral containers run in this poc an existing pod to perform user-initiated ac cannot be specified when creating a pod, a the pod spec. In order to add an ephemera pod's ephemeralcontainers subresource.
`hostAliases`	`array`	HostAliases is an optional list of hosts and hosts file if specified.
`hostIPC`	`boolean`	Use the host's ipc namespace. Optional: D
`hostNetwork`	`boolean`	Host networking requested for this pod. Us using HostNetwork you should specify port `hostNetwork` is true, specified `hostPor` `containerPort`, and unspecified `hostP` defaulted to match `containerPort`. Defa
`hostPID`	`boolean`	Use the host's pid namespace. Optional: D

Property	Type	Description
`hostUsers`	`boolean`	Use the host's user namespace. Optional: present, the pod will be run in the host use needs a feature only available to the host u kernel module with CAP_SYS_MODULE. V created for the pod. Setting false is useful f vulnerabilities even allowing users to run th having root privileges on the host. This fiel servers that enable the UserNamespacesS
`hostname`	`string`	Specifies the hostname of the Pod If not sp to a system-defined value.
`hostnameOverride`	`string`	HostnameOverride specifies an explicit ove perceived by the pod. This field only specif affect its DNS records. When this field is se • It takes precedence over the values set • The Pod's hostname will be set to this v • `setHostnameAsFQDN` must be nil or se • `hostNetwork` must be set to false. This field must be a valid DNS subdomain most 64 characters. Requires the Hostnam

Property	Type	Description
`imagePullSecrets`	`array`	ImagePullSecrets is an optional list of refer namespace to use for pulling any of the im these secrets will be passed to individual p More info: https://kubernetes.io/docs/conce imagepullsecrets-on-a-pod ↗
`initContainers`	`array`	List of initialization containers belonging to order prior to containers being started. If ar considered to have failed and is handled a for an init container or normal container mu containers may not have Lifecycle actions, Startup probes. The resourceRequirement account during scheduling by finding the hi type, and then using the max of that value Limits are applied to init containers in a sim currently be added or removed. Cannot be https://kubernetes.io/docs/concepts/worklo
`nodeName`	`string`	NodeName indicates in which node this po candidate for scheduling by the scheduler field is set, the kubelet for this node becom pod. This field should not be used to expre on a specific node. https://kubernetes.io/dc pod-node/#nodename ↗
`nodeSelector`	`object`	NodeSelector is a selector which must be t which must match a node's labels for the p info: https://kubernetes.io/docs/concepts/c

Property	Type	Description
`os`	`object`	Specifies the OS of the containers in the p restricted if this is set. If the OS field is set to linux, the following f securityContext.windowsOptions If the OS field is set to windows, following f • spec.hostPID • spec.hostIPC • spec.hostUsers • spec.resources • spec.securityContext.appArmorProfile • spec.securityContext.seLinuxOptions • spec.securityContext.seccompProfile • spec.securityContext.fsGroup • spec.securityContext.fsGroupChangeP • spec.securityContext.sysctls • spec.shareProcessNamespace • spec.securityContext.runAsUser • spec.securityContext.runAsGroup • spec.securityContext.supplementalGrou • spec.securityContext.supplementalGrou • spec.containers[*].securityContext.appA • spec.containers[*].securityContext.seLi • spec.containers[*].securityContext.secc • spec.containers[*].securityContext.capa • spec.containers[*].securityContext.read • spec.containers[*].securityContext.privil

Property	Type	Description
		• spec.containers[*].securityContext.allov • spec.containers[*].securityContext.proc • spec.containers[*].securityContext.runA • spec.containers[*].securityContext.runA
`overhead`	`object`	Overhead represents the resource overhea given RuntimeClass. This field will be auto RuntimeClass admission controller. If the F enabled, overhead must not be set in Pod admission controller will reject Pod create already set. If RuntimeClass is configured will be set to the value defined in the corre remain unset and treated as zero. More inf https://git.k8s.io/enhancements/keps/sig-n
`preemptionPolicy`	`string`	PreemptionPolicy is the Policy for preempt Never, PreemptLowerPriority. Defaults to F
`priority`	`integer`	The priority value. Various system compon the pod. When Priority Admission Controlle setting this field. The admission controller PriorityClassName. The higher the value, t
`priorityClassName`	`string`	If specified, indicates the pod's priority. "sy critical" are two special keywords which inc former being the highest priority. Any other PriorityClass object with that name. If not s or zero if there is no default.

Property	Type	Description
`readinessGates`	`array`	If specified, all readiness gates will be eval when all its containers are ready AND all c gates have status equal to "True" More inf https://git.k8s.io/enhancements/keps/sig-n
`resourceClaims`	`array`	ResourceClaims defines which ResourceC before the Pod is allowed to start. The res containers which consume them by name. This is a stable field but requires that the D is enabled. This field is immutable.
`resources`	`object`	Resources is the total amount of CPU and containers in the pod. It supports specifyin "memory" and "hugepages-" resource nam supported. This field enables fine-grained control over allowing resource sharing among containe This is an alpha field and requires enablin
`restartPolicy`	`string`	Restart policy for all containers within the p some contexts, only a subset of those valu More info: https://kubernetes.io/docs/conce lifecycle/#restart-policy ↗

Property	Type	Description
`runtimeClassName`	`string`	RuntimeClassName refers to a RuntimeCl which should be used to run this pod. If no named class, the pod will not be run. If uns will be used, which is an implicit class with default runtime handler. More info: https://g node/585-runtime-class ↗
`schedulerName`	`string`	If specified, the pod will be dispatched by s pod will be dispatched by default schedule
`schedulingGates`	`array`	SchedulingGates is an opaque list of value the pod. If schedulingGates is not empty, tl state and the scheduler will not attempt to : SchedulingGates can only be set at pod cr afterwards.
`securityContext`	`object`	SecurityContext holds pod-level security at settings. Optional: Defaults to empty. See t each field.
`serviceAccount`	`string`	DeprecatedServiceAccount is a deprecate Deprecated: Use serviceAccountName ins

Property	Type	Description
`serviceAccountName`	`string`	ServiceAccountName is the name of the S More info: https://kubernetes.io/docs/tasks/ service-account/ ↗
`setHostnameAsFQDN`	`boolean`	If true the pod's hostname will be configure leaf name (the default). In Linux containers hostname field of the kernel (the nodenam containers, this means setting the registry HKEY_LOCAL_MACHINE\SYSTEM\Curre to FQDN. If a pod does not have FQDN, th
`shareProcessNamespace`	`boolean`	Share a single process namespace betwee this is set containers will be able to view ar containers in the same pod, and the first pr assigned PID 1. HostPID and ShareProces Optional: Default to false.
`subdomain`	`string`	If specified, the fully qualified Pod hostnam pod will not have a domainname at all.

Property	Type	Description
`terminationGracePeriodSeconds`	`integer`	Optional duration in seconds the pod need decreased in delete request. Value must b indicates stop immediately via the kill signa value is nil, the default grace period will be duration in seconds after the processes ru signal and the time when the processes ar this value longer than the expected cleanu seconds.
`tolerations`	`array`	If specified, the pod's tolerations.
`topologySpreadConstraints`	`array`	TopologySpreadConstraints describes how across topology domains. Scheduler will s the constraints. All topologySpreadConstra
`volumes`	`array`	List of volumes that can be mounted by co info: https://kubernetes.io/docs/concepts/st
`workloadRef`	`object`	WorkloadRef provides a reference to the W This field is used by the scheduler to identi group scheduling policies. The Workload o exist at the time the Pod is created. This fie with the same name may be recreated wit scheduling may result in the placement no

.spec.resources.kubernetesResource.spec.template.spec.affinity

Description

If specified, the pod's scheduling constraints

Type

`object`

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

.spec.resources.kubernetesResource.spec.template.spec.affinity.nodeAffinity

Description

Describes node affinity scheduling rules for the pod.

Type

`object`

Property	Type	Description
`preferredDuringSchedulingIgnoredDuringExecution`	`array`	The scheduler will pr to schedule pods to nodes that satisfy the affinity expressions specified by this field, it may choose a node violates one or more the expressions. The node that is most preferred is the one v the greatest sum of weights, i.e. for each node that meets all o scheduling requireme (resource request, requiredDuringSched affinity expressions, e compute a sum by iterating through the elements of this field adding "weight" to the sum if the node matc the corresponding matchExpressions; th node(s) with the high sum are the most preferred.
`requiredDuringSchedulingIgnoredDuringExecution`	`object`	If the affinity requirem specified by this field not met at scheduling time, the pod will not scheduled onto the n

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

.spec.resources.kubernetesResource.spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

Description

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

Type

`object`

Required

`preference` `weight`

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

.spec.resources.kubernetesResource.spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference

Description

A node selector term, associated with the corresponding weight.

Type

`object`

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

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

.spec.resources.kubernetesResource.spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[]

Description

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

Type

`object`

Required

`key` `operator`

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

Property	Type	Description
`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.resources.kubernetesResource.spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchExpressions[].values[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[]

Description

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

Type

`object`

Required

`key` `operator`

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

Property	Type	Description
`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.resources.kubernetesResource.spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matchFields[].values[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDur

ingExecution

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.resources.kubernetesResource.spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[]

Description

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

Type

`object`

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

.spec.resources.kubernetesResource.spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[]

Description

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

Type

`object`

Required

`key` `operator`

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

.spec.resources.kubernetesResource.spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchExpressions[].values[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[]

Description

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

Type

`object`

Required

`key` `operator`

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

.spec.resources.kubernetesResource.spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms[].matchFields[].values

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDur

ingExecution.nodeSelectorTerms[].matchFields[].values[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAffinity

Description

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

Type

`object`

Property	Type	Description
`preferredDuringSchedulingIgnoredDuringExecution`	`array`	The scheduler will pr to schedule pods to nodes that satisfy the affinity expressions specified by this field, it may choose a node violates one or more the expressions. The node that is most preferred is the one v the greatest sum of weights, i.e. for each node that meets all o scheduling requireme (resource request, requiredDuringSched affinity expressions, e compute a sum by

Property	Type	Description
		iterating through the elements of this field adding "weight" to the sum if the node has p which matches the corresponding podAffinityTerm; the node(s) with the high sum are the most preferred.
`requiredDuringSchedulingIgnoredDuringExecution`	`array`	If the affinity requirem specified by this field not met at scheduling time, the pod will not scheduled onto the n If the affinity requirem specified by this field cease to be met at sc point during pod execution (e.g. due to pod label update), the system may or may n try to eventually evict pod from its node. Wl there are multiple elements, the lists of nodes corresponding each podAffinityTerm intersected, i.e. all ter must be satisfied.

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

Description

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

Type

`object`

Required

`podAffinityTerm` `weight`

Property	Type	Description
`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.resources.kubernetesResource.spec.template.spec.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
		`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.
`mismatchLabelKeys`	`array`	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `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.
`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.

Property	Type	Description
`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.resources.kubernetesResource.spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector

Description

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

Type

`object`

Property	Type	Description
`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.resources.kubernetesResource.spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]

Description

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

Type

`object`

Required

`key` `operator`

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

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels

Description

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

Type

`object`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector

Description

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

Type

`object`

Property	Type	Description
`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.resources.kubernetesResource.spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]

Description

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

Type

`object`

Required

`key` `operator`

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

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchLabels

Description

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

Type

`object`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.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.resources.kubernetesResource.spec.template.spec.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) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty.

Property	Type	Description
		The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set.
`mismatchLabelKeys`	`array`	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `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.
`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

Property	Type	Description
		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.resources.kubernetesResource.spec.template.spec.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

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

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.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.

Property	Type	Description
`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.resources.kubernetesResource.spec.template.spec.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.resources.kubernetesResource.spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.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.resources.kubernetesResource.spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.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
`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.resources.kubernetesResource.spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.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.resources.kubernetesResource.spec.template.spec.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.resources.kubernetesResource.spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.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.resources.kubernetesResource.spec.template.spec.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.resources.kubernetesResource.spec.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity

Description

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

Type

`object`

Property	Type	Description
`preferredDuringSchedulingIgnoredDuringExecution`	`array`	The scheduler will pr to schedule pods to nodes that satisfy the anti-affinity expressio specified by this field, it may choose a node violates one or more the expressions. The node that is most preferred is the one v the greatest sum of weights, i.e. for each node that meets all o

Property	Type	Description
		scheduling requireme (resource request, requiredDuringSched anti-affinity expressio etc.), compute a sum iterating through the elements of this field subtracting "weight" f the sum if the node h pods which matches corresponding podAffinityTerm; the node(s) with the high sum are the most preferred.
`requiredDuringSchedulingIgnoredDuringExecution`	`array`	If the anti-affinity requirements specifie this field are not met scheduling time, the p will not be scheduled the node. If the anti-affinity requirements specified by this field cease to be met at sc point during pod execution (e.g. due to pod label update), the system may or may n try to eventually evict pod from its node. Wh there are multiple elements, the lists of nodes corresponding

Property	Type	Description
		each podAffinityTerm intersected, i.e. all ter must be satisfied.

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling anti-affinity expressions, etc.), compute a sum by iterating through the elements of this field and subtracting "weight" from 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.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[]

Description

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

Type

`object`

Required

`podAffinityTerm` `weight`

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

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm

Description

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

Type

`object`

Required

`topologyKey`

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

Property	Type	Description
		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.
`mismatchLabelKeys`	`array`	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with `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.
`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

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

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector

Description

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

Type

`object`

Property	Type	Description
`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.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[]

Description

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

Type

`object`

Required

`key` `operator`

Property	Type	Description
`key`	`string`	key is the label key that the selector applies to.
`operator`	`string`	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
`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.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchExpressions[].values[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.labelSelector.matchLabels

Description

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

Type

`object`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys

Description

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

empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.matchLabelKeys[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.mismatchLabelKeys[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector

Description

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

Type

`object`

Property	Type	Description
`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.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnored

DuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[]

Description

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

Type

`object`

Required

`key` `operator`

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

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.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaceSelector.matchExpressions[].values[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnored

DuringExecution[].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.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm.namespaces[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnored

DuringExecution

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.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[]

Description

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

Type

`object`

Required

`topologyKey`

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

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

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

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector

Description

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

Type

`object`

Property	Type	Description
`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.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[]

Description

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

Type

`object`

Required

`key` `operator`

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

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchExpressions[].values[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].labelSelector.matchLabels

Description

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

Type

`object`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].matchLabelKeys[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].mismatchLabelKeys[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector

Description

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

Type

`object`

Property	Type	Description
`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.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[]

Description

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

Type

`object`

Required

`key` `operator`

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

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchExpressions[].values[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaceSelector.matchLabels

Description

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

Type

`object`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[].namespaces[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.containers

Description

List of containers belonging to the pod. Containers cannot currently be added or removed. There must be at least one container in a Pod. Cannot be updated.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.containers[]

Description

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

Type

`object`

Required

`name`

Property	Type	Description
`args`	`array`	Arguments to the entrypoint. The container imag if this is not provided. Variable references $(VAR expanded using the container's environment. If a be resolved, the reference in the input string will Double $$ are reduced to a single $, which allow the $(VAR_NAME) syntax: i.e. "$$(VAR_NAME)' string literal "$(VAR_NAME)". Escaped reference expanded, regardless of whether the variable ex Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-appli command-argument-container/#running-a-comm

Property	Type	Description
`command`	`array`	Entrypoint array. Not executed within a shell. The image's ENTRYPOINT is used if this is not provi references $(VAR_NAME) are expanded using tl environment. If a variable cannot be resolved, th the input string will be unchanged. Double $$ are single $, which allows for escaping the $(VAR_N i.e. "$$(VAR_NAME)" will produce the string liter "$(VAR_NAME)". Escaped references will never regardless of whether the variable exists or not. updated. More info: https://kubernetes.io/docs/ta application/define-command-argument-container command-in-a-shell ↗
`env`	`array`	List of environment variables to set in the contair updated.
`envFrom`	`array`	List of sources to populate environment variables container. The keys defined within a source may printable ASCII characters except '='. When a ke multiple sources, the value associated with the la take precedence. Values defined by an Env with will take precedence. Cannot be updated.
`image`	`string`	Container image name. More info: https://kubernetes.io/docs/concepts/containers/ir field is optional to allow higher level config mana default or override container images in workload Deployments and StatefulSets.

Property	Type	Description
`imagePullPolicy`	`string`	Image pull policy. One of Always, Never, IfNotPre to Always if :latest tag is specified, or IfNotPrese Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/ir images ↗
`lifecycle`	`object`	Actions that the management system should tak container lifecycle events. Cannot be updated.
`livenessProbe`	`object`	Periodic probe of container liveness. Container v the probe fails. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/p lifecycle#container-probes ↗
`name`	`string`	Name of the container specified as a DNS_LABE container in a pod must have a unique name (DN Cannot be updated.
`ports`	`array`	List of ports to expose from the container. Not sp here DOES NOT prevent that port from being ex which is listening on the default "0.0.0.0" addres container will be accessible from the network. M array with strategic merge patch may corrupt the information See https://github.com/kubernetes/kubernetes/issues Cannot be updated.

Property	Type	Description
`readinessProbe`	`object`	Periodic probe of container service readiness. C removed from service endpoints if the probe fails updated. More info: https://kubernetes.io/docs/concepts/workloads/p lifecycle#container-probes ↗
`resizePolicy`	`array`	Resources resize policy for the container. This fi set on ephemeral containers.
`resources`	`object`	Compute Resources required by this container. updated. More info: https://kubernetes.io/docs/concepts/configuratio resources-containers/ ↗
`restartPolicy`	`string`	RestartPolicy defines the restart behavior of indi in a pod. This overrides the pod-level restart poli field is not specified, the restart behavior is defin restart policy and the container type. Additionally RestartPolicy as "Always" for the init container w following effect: this init container will be continu exit until all regular containers have terminated. containers have completed, all init containers wit "Always" will be shut down. This lifecycle differs containers and is often referred to as a "sidecar" Although this init container still starts in the init c sequence, it does not wait for the container to co proceeding to the next init container. Instead, the container starts immediately after this init contai after any startupProbe has successfully complet

Property	Type	Description
`restartPolicyRules`	`array`	Represents a list of rules to be checked to deterr container should be restarted on exit. The rules a order. Once a rule matches a container exit cond remaining rules are ignored. If no rule matches tl condition, the Container-level restart policy deter whether the container is restarted or not. Constra rules: • At most 20 rules are allowed. • Rules can have the same action. • Identical rules are not forbidden in validations are specified, container MUST set RestartPol even it if matches the Pod's RestartPolicy.
`securityContext`	`object`	SecurityContext defines the security options the be run with. If set, the fields of SecurityContext o equivalent fields of PodSecurityContext. More inf https://kubernetes.io/docs/tasks/configure-pod-container/security-context/ ↗
`startupProbe`	`object`	StartupProbe indicates that the Pod has success If specified, no other probes are executed until th successfully. If this probe fails, the Pod will be re the livenessProbe failed. This can be used to pro probe parameters at the beginning of a Pod's life might take a long time to load data or warm a ca steady-state operation. This cannot be updated. https://kubernetes.io/docs/concepts/workloads/p lifecycle#container-probes ↗

Property	Type	Description
`stdin`	`boolean`	Whether this container should allocate a buffer fo container runtime. If this is not set, reads from st container will always result in EOF. Default is fals
`stdinOnce`	`boolean`	Whether the container runtime should close the s after it has been opened by a single attach. Whe the stdin stream will remain open across multiple sessions. If stdinOnce is set to true, stdin is oper start, is empty until the first client attaches to stdi remains open and accepts data until the client di which time stdin is closed and remains closed ur is restarted. If this flag is false, a container proce from stdin will never receive an EOF. Default is fa
`terminationMessagePath`	`string`	Optional: Path at which the file to which the conta termination message will be written is mounted in container's filesystem. Message written is intend final status, such as an assertion failure messag truncated by the node if greater than 4096 bytes. message length across all containers will be limit Defaults to /dev/termination-log. Cannot be upda
`terminationMessagePolicy`	`string`	Indicate how the termination message should be will use the contents of terminationMessagePath container status message on both success and f FallbackToLogsOnError will use the last chunk o output if the termination message file is empty ar exited with an error. The log output is limited to 2 lines, whichever is smaller. Defaults to File. Cann

Property	Type	Description
`tty`	`boolean`	Whether this container should allocate a TTY for requires 'stdin' to be true. Default is false.
`volumeDevices`	`array`	volumeDevices is the list of block devices to be u container.
`volumeMounts`	`array`	Pod volumes to mount into the container's filesys updated.
`workingDir`	`string`	Container's working directory. If not specified, the runtime's default will be used, which might be co container image. Cannot be updated.

.spec.resources.kubernetesResource.spec.template.spec.containers[].args

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.containers[].args[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.containers[].command

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.containers[].command[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.containers[].env

Description

List of environment variables to set in the container. Cannot be updated.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.containers[].env[]

Description

EnvVar represents an environment variable present in a Container.

Type

`object`

Required

`name`

Property	Type	Description
`name`	`string`	Name of the environment variable. May consist of any printable ASCII characters except '='.
`value`	`string`	Variable references $(VAR_NAME) are expanded using the previously defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input string will be unchanged. Double $$ are reduced to a single $, which allows for escaping the $(VAR_NAME) syntax: i.e. "$$(VAR_NAME)" will produce the string literal "$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".

Property	Type	Description
`valueFrom`	`object`	Source for the environment variable's value. Cannot be used if value is not empty.

.spec.resources.kubernetesResource.spec.template.spec.containers[].env[].valueFrom

Description

Source for the environment variable's value. Cannot be used if value is not empty.

Type

`object`

Property	Type	Description
`configMapKeyRef`	`object`	Selects a key of a ConfigMap.
`fieldRef`	`object`	Selects a field of the pod: supports metadata.name, metadata.namespace, `metadata.labels['<KEY>']`, `metadata.annotations['<KEY>']`, spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP, status.podIPs.
`fileKeyRef`	`object`	FileKeyRef selects a key of the env file. Requires the EnvFiles feature gate to be enabled.

Property	Type	Description
`resourceFieldRef`	`object`	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.
`secretKeyRef`	`object`	Selects a key of a secret in the pod's namespace

.spec.resources.kubernetesResource.spec.template.spec.containers[].env[].valueFrom.configMapKeyRef

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

Property	Type	Description
`key`	`string`	The key to select.
`name`	`string`	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info:

Property	Type	Description
		https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`optional`	`boolean`	Specify whether the ConfigMap or its key must be defined

.spec.resources.kubernetesResource.spec.template.spec.containers[].env[].valueFrom.fieldRef

Description

Selects a field of the pod: supports metadata.name, metadata.namespace, `metadata.labels['<KEY>']`, `metadata.annotations['<KEY>']`, spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP, status.podIPs.

Type

`object`

Required

`fieldPath`

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

.spec.resources.kubernetesResource.spec.template.spec.containers[].env[].valueFrom.fileKeyRef

Description

FileKeyRef selects a key of the env file. Requires the EnvFiles feature gate to be enabled.

Type

`object`

Required

`key` `path` `volumeName`

Property	Type	Description
`key`	`string`	The key within the env file. An invalid key will prevent the pod from starting. The keys defined within a source may consist of any printable ASCII characters except '='. During Alpha stage of the EnvFiles feature gate, the key size is limited to 128 characters.
`optional`	`boolean`	Specify whether the file or its key must be defined. If the file or key does not exist, then the env var is not published. If optional is set to true and the specified key does not exist, the environment variable will not be set in the Pod's containers. If optional is set to false and the specified key does not exist, an error will be returned during Pod creation.
`path`	`string`	The path within the volume from which to select the file. Must be relative and may not contain the '..' path or start with '..'.
`volumeName`	`string`	The name of the volume mount containing the env file.

.spec.resources.kubernetesResource.spec.template.spec.containers[].env[].valueFrom.resourceFieldRef

Description

Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.

Type

`object`

Required

`resource`

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

.spec.resources.kubernetesResource.spec.template.spec.containers[].env[].valueFrom.secretKeyRef

Description

Selects a key of a secret in the pod's namespace

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. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`optional`	`boolean`	Specify whether the Secret or its key must be defined

.spec.resources.kubernetesResource.spec.template.spec.containers[].envFrom

Description

List of sources to populate environment variables in the container. The keys defined within a source may consist of any printable ASCII characters except '='. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.containers[].envFrom[]

Description

EnvFromSource represents the source of a set of ConfigMaps or Secrets

Type

`object`

Property	Type	Description
`configMapRef`	`object`	The ConfigMap to select from
`prefix`	`string`	Optional text to prepend to the name of each environment variable. May consist of any printable ASCII characters except '='.
`secretRef`	`object`	The Secret to select from

.spec.resources.kubernetesResource.spec.template.spec.containers[].envFrom[].configMapRef

Description

The ConfigMap to select from

Type

`object`

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

.spec.resources.kubernetesResource.spec.template.spec.containers[].envFrom[].secretRef

Description

The Secret to select from

Type

`object`

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

.spec.resources.kubernetesResource.spec.template.spec.containers[].lifecycle

Description

Actions that the management system should take in response to container lifecycle events. Cannot be updated.

Type

`object`

Property	Type	Description
`postStart`	`object`	PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks ↗
`preStop`	`object`	PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks ↗

Property	Type	Description
`stopSignal`	`string`	StopSignal defines which signal will be sent to a container when it is being stopped. If not specified, the default is defined by the container runtime in use. StopSignal can only be set for Pods with a non-empty .spec.os.name

.spec.resources.kubernetesResource.spec.template.spec.containers[].lifecycle.postStart

Description

PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks

Type

`object`

Property	Type	Description
`exec`	`object`	Exec specifies a command to execute in the container.
`httpGet`	`object`	HTTPGet specifies an HTTP GET request to perform.
`sleep`	`object`	Sleep represents a duration that the container should sleep.
`tcpSocket`	`object`	Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for backward compatibility.

Property	Type	Description
		There is no validation of this field and lifecycle hooks will fail at runtime when it is specified.

.spec.resources.kubernetesResource.spec.template.spec.containers[].lifecycle.postStart.exec

Description

Exec specifies a command to execute in the container.

Type

`object`

Property	Type	Description
`command`	`array`	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.resources.kubernetesResource.spec.template.spec.containers[].lifecycle.postStart.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.containers[].lifecycle.postStart.exec.command[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.containers[].lifecycle.postStart.httpGet

Description

HTTPGet specifies an HTTP GET request to perform.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
`httpHeaders`	`array`	Custom headers to set in the request. HTTP allows repeated headers.
`path`	`string`	Path to access on the HTTP server.

Property	Type	Description
`port`		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
`scheme`	`string`	Scheme to use for connecting to the host. Defaults to HTTP.

.spec.resources.kubernetesResource.spec.template.spec.containers[].lifecycle.postStart.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.containers[].lifecycle.postStart.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
`value`	`string`	The header field value

.spec.resources.kubernetesResource.spec.template.spec.containers[].lifecycle.postStart.sleep

Description

Sleep represents a duration that the container should sleep.

Type

`object`

Required

`seconds`

Property	Type	Description
`seconds`	`integer`	Seconds is the number of seconds to sleep.

.spec.resources.kubernetesResource.spec.template.spec.containers[].lifecycle.postStart.tcpSocket

Description

Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for backward compatibility. There is no validation of this field and lifecycle hooks will fail at runtime when it

is specified.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Optional: Host name to connect to, defaults to the pod IP.
`port`		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

.spec.resources.kubernetesResource.spec.template.spec.containers[].lifecycle.preStop

Description

PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks

Type

`object`

Property	Type	Description
`exec`	`object`	Exec specifies a command to execute in the container.
`httpGet`	`object`	HTTPGet specifies an HTTP GET request to perform.
`sleep`	`object`	Sleep represents a duration that the container should sleep.
`tcpSocket`	`object`	Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for backward compatibility. There is no validation of this field and lifecycle hooks will fail at runtime when it is specified.

.spec.resources.kubernetesResource.spec.template.spec.containers[].lifecycle.preStop.exec

Description

Exec specifies a command to execute in the container.

Type

`object`

Property	Type	Description
`command`	`array`	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to

Property	Type	Description
		explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.resources.kubernetesResource.spec.template.spec.containers[].lifecycle.preStop.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.containers[].lifecycle.preStop.exec.command[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.containers[].lifecycle.preStop.httpGet

Description

HTTPGet specifies an HTTP GET request to perform.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
`httpHeaders`	`array`	Custom headers to set in the request. HTTP allows repeated headers.
`path`	`string`	Path to access on the HTTP server.
`port`		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
`scheme`	`string`	Scheme to use for connecting to the host. Defaults to HTTP.

.spec.resources.kubernetesResource.spec.template.spec.containers[].lifecycle.preStop.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.containers[].lifecycle.preStop.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
`value`	`string`	The header field value

.spec.resources.kubernetesResource.spec.template.spec.containers[].lifecycle.preStop.sleep

Description

Sleep represents a duration that the container should sleep.

Type

`object`

Required

`seconds`

Property	Type	Description
`seconds`	`integer`	Seconds is the number of seconds to sleep.

.spec.resources.kubernetesResource.spec.template.spec.containers[].lifecycle.preStop.tcpSocket

Description

Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for backward compatibility. There is no validation of this field and lifecycle hooks will fail at runtime when it is specified.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Optional: Host name to connect to, defaults to the pod IP.
`port`		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

.spec.resources.kubernetesResource.spec.template.spec.containers[].livenessProbe

Description

Periodic probe of container liveness. Container will be restarted if the probe fails. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

Type

`object`

Property	Type	Description
`exec`	`object`	Exec specifies a command to execute in th
`failureThreshold`	`integer`	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
`grpc`	`object`	GRPC specifies a GRPC HealthCheckReq
`httpGet`	`object`	HTTPGet specifies an HTTP GET request
`initialDelaySeconds`	`integer`	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗
`periodSeconds`	`integer`	How often (in seconds) to perform the prob 10 seconds. Minimum value is 1.
`successThreshold`	`integer`	Minimum consecutive successes for the pr considered successful after having failed. I

Property	Type	Description
		Must be 1 for liveness and startup. Minimu
`tcpSocket`	`object`	TCPSocket specifies a connection to a TC
`terminationGracePeriodSeconds`	`integer`	Optional duration in seconds the pod need gracefully upon probe failure. The grace pe duration in seconds after the processes ru pod are sent a termination signal and the ti processes are forcibly halted with a kill sig value longer than the expected cleanup tim process. If this value is nil, the pod's terminationGracePeriodSeconds will be us Otherwise, this value overrides the value p pod spec. Value must be non-negative inte value zero indicates stop immediately via t (no opportunity to shut down). This is a bet requires enabling ProbeTerminationGraceP gate. Minimum value is 1. spec.terminationGracePeriodSeconds is u
`timeoutSeconds`	`integer`	Number of seconds after which the probe t Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗

.spec.resources.kubernetesResource.spec.template.spec.containers[].livenessProbe.exec

Description

Exec specifies a command to execute in the container.

Type

`object`

Property	Type	Description
`command`	`array`	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.resources.kubernetesResource.spec.template.spec.containers[].livenessProbe.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.containers[].livenessProbe.exec.command[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.containers[].livenessProbe.grpc

Description

GRPC specifies a GRPC HealthCheckRequest.

Type

`object`

Required

`port`

Property	Type	Description
`port`	`integer`	Port number of the gRPC service. Number must be in the range 1 to 65535.
`service`	`string`	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md↗). If this is not specified, the default behavior is defined by gRPC.

.spec.resources.kubernetesResource.spec.template.spec.containers[].livenessProbe.httpGet

Description

HTTPGet specifies an HTTP GET request to perform.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
`httpHeaders`	`array`	Custom headers to set in the request. HTTP allows repeated headers.
`path`	`string`	Path to access on the HTTP server.
`port`		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
`scheme`	`string`	Scheme to use for connecting to the host. Defaults to HTTP.

.spec.resources.kubernetesResource.spec.template.spec.containers[].livenessProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.containers[].livenessProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
`value`	`string`	The header field value

.spec.resources.kubernetesResource.spec.template.spec.containers[].livenessProbe.tcpSocket

Description

TCPSocket specifies a connection to a TCP port.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Optional: Host name to connect to, defaults to the pod IP.
`port`		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

.spec.resources.kubernetesResource.spec.template.spec.containers[].ports

Description

List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address inside a container will be accessible from the network. Modifying this array with strategic merge patch may corrupt the data. For more information See https://github.com/kubernetes/kubernetes/issues/108255. Cannot be updated.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.containers[].ports[]

Description

ContainerPort represents a network port in a single container.

Type

`object`

Required

`containerPort`

Property	Type	Description
`containerPort`	`integer`	Number of port to expose on the pod's IP address. This must be a valid port number, 0 < x < 65536.
`hostIP`	`string`	What host IP to bind the external port to.
`hostPort`	`integer`	Number of port to expose on the host. If specified, this must be a valid port number, 0 < x < 65536. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this.
`name`	`string`	If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services.
`protocol`	`string`	Protocol for port. Must be UDP, TCP, or SCTP. Defaults to "TCP".

.spec.resources.kubernetesResource.spec.template.spec.containers[].readinessProbe

Description

Periodic probe of container service readiness. Container will be removed from service endpoints if the probe fails. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

Type

`object`

Property	Type	Description
`exec`	`object`	Exec specifies a command to execute in th
`failureThreshold`	`integer`	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
`grpc`	`object`	GRPC specifies a GRPC HealthCheckReq
`httpGet`	`object`	HTTPGet specifies an HTTP GET request
`initialDelaySeconds`	`integer`	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗
`periodSeconds`	`integer`	How often (in seconds) to perform the prob 10 seconds. Minimum value is 1.
`successThreshold`	`integer`	Minimum consecutive successes for the pr considered successful after having failed. I Must be 1 for liveness and startup. Minimu

Property	Type	Description
`tcpSocket`	`object`	TCPSocket specifies a connection to a TC
`terminationGracePeriodSeconds`	`integer`	Optional duration in seconds the pod need gracefully upon probe failure. The grace pe duration in seconds after the processes ru pod are sent a termination signal and the ti processes are forcibly halted with a kill sig value longer than the expected cleanup tim process. If this value is nil, the pod's terminationGracePeriodSeconds will be us Otherwise, this value overrides the value p pod spec. Value must be non-negative inte value zero indicates stop immediately via t (no opportunity to shut down). This is a bet requires enabling ProbeTerminationGraceP gate. Minimum value is 1. spec.terminationGracePeriodSeconds is u
`timeoutSeconds`	`integer`	Number of seconds after which the probe t Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗

.spec.resources.kubernetesResource.spec.template.spec.containers[].readinessProbe.exec

Description

Exec specifies a command to execute in the container.

Type

`object`

Property	Type	Description
`command`	`array`	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.resources.kubernetesResource.spec.template.spec.containers[].readinessProbe.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.containers[].readinessProbe.exec.command[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.containers[].readinessProbe.grpc

Description

GRPC specifies a GRPC HealthCheckRequest.

Type

`object`

Required

`port`

Property	Type	Description
`port`	`integer`	Port number of the gRPC service. Number must be in the range 1 to 65535.
`service`	`string`	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md↗). If this is not specified, the default behavior is defined by gRPC.

.spec.resources.kubernetesResource.spec.template.spec.containers[].readinessProbe.httpGet

Description

HTTPGet specifies an HTTP GET request to perform.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
`httpHeaders`	`array`	Custom headers to set in the request. HTTP allows repeated headers.
`path`	`string`	Path to access on the HTTP server.
`port`		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
`scheme`	`string`	Scheme to use for connecting to the host. Defaults to HTTP.

.spec.resources.kubernetesResource.spec.template.spec.containers[].readinessProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.containers[].readinessProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
`value`	`string`	The header field value

.spec.resources.kubernetesResource.spec.template.spec.containers[].readinessProbe.tcpSocket

Description

TCPSocket specifies a connection to a TCP port.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Optional: Host name to connect to, defaults to the pod IP.
`port`		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

.spec.resources.kubernetesResource.spec.template.spec.containers[].resizePolicy

Description

Resources resize policy for the container. This field cannot be set on ephemeral containers.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.containers[].resizePolicy[]

Description

ContainerResizePolicy represents resource resize policy for the container.

Type

`object`

Required

`resourceName` `restartPolicy`

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

.spec.resources.kubernetesResource.spec.template.spec.containers[].resources

Description

Compute Resources required by this container. Cannot be updated. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

Type

`object`

Property	Type	Description
`claims`	`array`	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This field depends on the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
`limits`	`object`	Limits describes the maximum amount of compute resources allowed. More info:

Property	Type	Description
		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.resources.kubernetesResource.spec.template.spec.containers[].resources.claims

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.containers[].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.resources.kubernetesResource.spec.template.spec.containers[].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.kubernetesResource.spec.template.spec.containers[].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.resources.kubernetesResource.spec.template.spec.containers[].restartPolicyRules

Description

Represents a list of rules to be checked to determine if the container should be restarted on exit. The rules are evaluated in order. Once a rule matches a container exit condition, the remaining rules are ignored. If no rule matches the container exit condition, the Container-level restart policy determines the whether the container is restarted or not. Constraints on the rules: - At most 20 rules are allowed. - Rules can have the same action. - Identical rules are not forbidden in validations. When rules are specified, container MUST set RestartPolicy explicitly even it if matches the Pod's RestartPolicy.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.containers[].restartPolicyRules[]

Description

ContainerRestartRule describes how a container exit is handled.

Type

`object`

Required

`action`

Property	Type	Description
`action`	`string`	Specifies the action taken on a container exit if the requirements are satisfied. The only possible value is "Restart" to restart the container.

Property	Type	Description
`exitCodes`	`object`	Represents the exit codes to check on container exits.

.spec.resources.kubernetesResource.spec.template.spec.containers[].restartPolicyRules[].exitCodes

Description

Represents the exit codes to check on container exits.

Type

`object`

Required

`operator`

Property	Type	Description
`operator`	`string`	Represents the relationship between the container exit code(s) and the specified values. Possible values are: • In: the requirement is satisfied if the container exit code is in the set of specified values. • NotIn: the requirement is satisfied if the container exit code is not in the set of specified values.
`values`	`array`	Specifies the set of values to check for container exit codes. At most 255 elements are allowed.

.spec.resources.kubernetesResource.spec.template.spec.containers[].restartPolicyRules[].exitCodes.values

Description

Specifies the set of values to check for container exit codes. At most 255 elements are allowed.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.containers[].restartPolicyRules[].exitCodes.values[]

Type

`integer`

.spec.resources.kubernetesResource.spec.template.spec.containers[].securityContext

Description

SecurityContext defines the security options the container should be run with. If set, the fields of SecurityContext override the equivalent fields of PodSecurityContext. More info: https://kubernetes.io/docs/tasks/configure-pod-container/security-context/

Type

`object`

Property	Type	Description
`allowPrivilegeEscalation`	`boolean`	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the no_new_privs flag will be set on the

Property	Type	Description
		container process. AllowPrivilegeEscalation is true always when the container is: 1. run as Privileged 2. has CAP_SYS_ADMIN Note that this field cannot be set when spec.os.name is windows.
`appArmorProfile`	`object`	appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when spec.os.name is windows.
`capabilities`	`object`	The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when spec.os.name is windows.
`privileged`	`boolean`	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.
`procMount`	`string`	procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked

Property	Type	Description
		paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows.
`readOnlyRootFilesystem`	`boolean`	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
`runAsGroup`	`integer`	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
`runAsNonRoot`	`boolean`	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

Property	Type	Description
`runAsUser`	`integer`	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
`seLinuxOptions`	`object`	The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
`seccompProfile`	`object`	The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.
`windowsOptions`	`object`	The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set

Property	Type	Description
		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.resources.kubernetesResource.spec.template.spec.containers[].securityContext.appArmorProfile

Description

appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Required

`type`

Property	Type	Description
`localhostProfile`	`string`	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
`type`	`string`	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container

Property	Type	Description
		runtime's default profile. Unconfined - no AppArmor enforcement.

.spec.resources.kubernetesResource.spec.template.spec.containers[].securityContext.capabilities

Description

The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Property	Type	Description
`add`	`array`	Added capabilities
`drop`	`array`	Removed capabilities

.spec.resources.kubernetesResource.spec.template.spec.containers[].securityContext.capabilities.add

Description

Added capabilities

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.containers[].securityContext.capabilities.add[]

Description

Capability represent POSIX capabilities type

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.containers[].securityContext.capabilities.drop

Description

Removed capabilities

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.containers[].securityContext.capabilities.drop[]

Description

Capability represent POSIX capabilities type

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.containers[].securityContext.seLinuxOptions

Description

The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value

specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

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

.spec.resources.kubernetesResource.spec.template.spec.containers[].securityContext.seccompProfile

Description

The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Required

`type`

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

.spec.resources.kubernetesResource.spec.template.spec.containers[].securityContext.windowsOptions

Description

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

Type

`object`

Property	Type	Description
`gmsaCredentialSpec`	`string`	GMSACredentialSpec is where the GMSA admission webhook

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

.spec.resources.kubernetesResource.spec.template.spec.containers[].startupProbe

Description

StartupProbe indicates that the Pod has successfully initialized. If specified, no other probes are executed until this completes successfully. If this probe fails, the Pod will be restarted, just as if the livenessProbe failed. This can be used to provide different probe parameters at the beginning of a Pod's lifecycle, when it might take a long time to load data or warm a cache, than during steady-state operation. This cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

Type

`object`

Property	Type	Description
`exec`	`object`	Exec specifies a command to execute in th
`failureThreshold`	`integer`	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
`grpc`	`object`	GRPC specifies a GRPC HealthCheckReq
`httpGet`	`object`	HTTPGet specifies an HTTP GET request
`initialDelaySeconds`	`integer`	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗

Property	Type	Description
`periodSeconds`	`integer`	How often (in seconds) to perform the prob 10 seconds. Minimum value is 1.
`successThreshold`	`integer`	Minimum consecutive successes for the pr considered successful after having failed. I Must be 1 for liveness and startup. Minimu
`tcpSocket`	`object`	TCPSocket specifies a connection to a TC
`terminationGracePeriodSeconds`	`integer`	Optional duration in seconds the pod need gracefully upon probe failure. The grace pe duration in seconds after the processes run pod are sent a termination signal and the ti processes are forcibly halted with a kill sign value longer than the expected cleanup tim process. If this value is nil, the pod's terminationGracePeriodSeconds will be us Otherwise, this value overrides the value p pod spec. Value must be non-negative inte value zero indicates stop immediately via t (no opportunity to shut down). This is a bet requires enabling ProbeTerminationGraceP gate. Minimum value is 1. spec.terminationGracePeriodSeconds is us

Property	Type	Description
`timeoutSeconds`	`integer`	Number of seconds after which the probe t Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗

.spec.resources.kubernetesResource.spec.template.spec.containers[].startupProbe.exec

Description

Exec specifies a command to execute in the container.

Type

`object`

Property	Type	Description
`command`	`array`	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.resources.kubernetesResource.spec.template.spec.containers[].startupProbe.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.containers[].startupProbe.exec.command[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.containers[].startupProbe.grpc

Description

GRPC specifies a GRPC HealthCheckRequest.

Type

`object`

Required

`port`

Property	Type	Description
`port`	`integer`	Port number of the gRPC service. Number must be in the range 1 to 65535.
`service`	`string`	Service is the name of the service to place in the gRPC HealthCheckRequest (see

Property	Type	Description
		https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

.spec.resources.kubernetesResource.spec.template.spec.containers[].startupProbe.httpGet

Description

HTTPGet specifies an HTTP GET request to perform.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
`httpHeaders`	`array`	Custom headers to set in the request. HTTP allows repeated headers.
`path`	`string`	Path to access on the HTTP server.

Property	Type	Description
`port`		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
`scheme`	`string`	Scheme to use for connecting to the host. Defaults to HTTP.

.spec.resources.kubernetesResource.spec.template.spec.containers[].startupProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.containers[].startupProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
`value`	`string`	The header field value

.spec.resources.kubernetesResource.spec.template.spec.containers[].startupProbe.tcpSocket

Description

TCPSocket specifies a connection to a TCP port.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Optional: Host name to connect to, defaults to the pod IP.
`port`		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

.spec.resources.kubernetesResource.spec.template.spec.containers[].volumeDevices

Description

volumeDevices is the list of block devices to be used by the container.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.containers[].volumeDevices[]

Description

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

Type

`object`

Required

`devicePath` `name`

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

.spec.resources.kubernetesResource.spec.template.spec.containers[].volumeMounts

Description

Pod volumes to mount into the container's filesystem. Cannot be updated.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.containers[].volumeMounts[]

Description

VolumeMount describes a mounting of a Volume within a container.

Type

`object`

Required

`mountPath` `name`

Property	Type	Description
`mountPath`	`string`	Path within the container at which the volume should be mounted. Must not contain ':'.
`mountPropagation`	`string`	mountPropagation determines how mounts are propagated from the host to container and the other way around. When not set, MountPropagationNone is used. This field is beta in 1.10. When RecursiveReadOnly is set to IfPossible or to Enabled, MountPropagation must be None or unspecified (which defaults to None).
`name`	`string`	This must match the Name of a Volume.

Property	Type	Description
`readOnly`	`boolean`	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.
`recursiveReadOnly`	`string`	RecursiveReadOnly specifies whether read-only mounts should be handled recursively. If ReadOnly is false, this field has no meaning and must be unspecified. If ReadOnly is true, and this field is set to Disabled, the mount is not made recursively read-only. If this field is set to IfPossible, the mount is made recursively read-only, if it is supported by the container runtime. If this field is set to Enabled, the mount is made recursively read-only if it is supported by the container runtime, otherwise the pod will not be started and an error will be generated to indicate the reason. If this field is set to IfPossible or Enabled, MountPropagation must be set to None (or be unspecified, which defaults to None). If this field is not specified, it is treated as an equivalent of Disabled.
`subPath`	`string`	Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).

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

.spec.resources.kubernetesResource.spec.template.spec.dnsConfig

Description

Specifies the DNS parameters of a pod. Parameters specified here will be merged to the generated DNS configuration based on DNSPolicy.

Type

`object`

Property	Type	Description
`nameservers`	`array`	A list of DNS name server IP addresses. This will be appended to the base nameservers generated from DNSPolicy. Duplicated nameservers will be removed.
`options`	`array`	A list of DNS resolver options. This will be merged with the base options generated from DNSPolicy. Duplicated entries will be removed. Resolution options given in Options will override those that appear in the base DNSPolicy.

Property	Type	Description
`searches`	`array`	A list of DNS search domains for host-name lookup. This will be appended to the base search paths generated from DNSPolicy. Duplicated search paths will be removed.

.spec.resources.kubernetesResource.spec.template.spec.dnsConfig.nameservers

Description

A list of DNS name server IP addresses. This will be appended to the base nameservers generated from DNSPolicy. Duplicated nameservers will be removed.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.dnsConfig.nameservers[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.dnsConfig.options

Description

A list of DNS resolver options. This will be merged with the base options generated from DNSPolicy. Duplicated entries will be removed. Resolution options given in Options will override those that appear in the base DNSPolicy.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.dnsConfig.options[]

Description

PodDNSConfigOption defines DNS resolver options of a pod.

Type

`object`

Property	Type	Description
`name`	`string`	Name is this DNS resolver option's name. Required.
`value`	`string`	Value is this DNS resolver option's value.

.spec.resources.kubernetesResource.spec.template.spec.dnsConfig.searches

Description

A list of DNS search domains for host-name lookup. This will be appended to the base search paths generated from DNSPolicy. Duplicated search paths will be removed.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.dnsConfig.searches[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers

Description

List of ephemeral containers run in this pod. Ephemeral containers may be run in an existing pod to perform user-initiated actions such as debugging. This list cannot be specified when creating a pod, and it cannot be modified by updating the pod spec. In order to add an ephemeral container to an existing pod, use the pod's ephemeralcontainers subresource.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[]

Description

An EphemeralContainer is a temporary container that you may add to an existing Pod for user-initiated activities such as debugging. Ephemeral containers have no resource or scheduling guarantees, and they will not be restarted when they exit or when a Pod is removed or restarted. The kubelet may evict a Pod if an ephemeral container causes the Pod to exceed its resource allocation. To add an ephemeral container, use the ephemeralcontainers subresource of an existing Pod. Ephemeral containers may not be removed or restarted.

Type

`object`

Required

`name`

Property	Type	Description
`args`	`array`	Arguments to the entrypoint. The image's CMD is not provided. Variable references $(VAR_NAME) using the container's environment. If a variable c

Property	Type	Description
		resolved, the reference in the input string will be Double $$ are reduced to a single $, which allow the $(VAR_NAME) syntax: i.e. "$$(VAR_NAME)' string literal "$(VAR_NAME)". Escaped reference expanded, regardless of whether the variable ex Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-appli command-argument-container/#running-a-comm
`command`	`array`	Entrypoint array. Not executed within a shell. The ENTRYPOINT is used if this is not provided. Vari $(VAR_NAME) are expanded using the containe If a variable cannot be resolved, the reference in will be unchanged. Double $$ are reduced to a s allows for escaping the $(VAR_NAME) syntax: i. "$$(VAR_NAME)" will produce the string literal "$ Escaped references will never be expanded, reg whether the variable exists or not. Cannot be up https://kubernetes.io/docs/tasks/inject-data-appli command-argument-container/#running-a-comm
`env`	`array`	List of environment variables to set in the contair updated.
`envFrom`	`array`	List of sources to populate environment variables container. The keys defined within a source may printable ASCII characters except '='. When a ke multiple sources, the value associated with the la take precedence. Values defined by an Env with will take precedence. Cannot be updated.

Property	Type	Description
`image`	`string`	Container image name. More info: https://kubernetes.io/docs/concepts/containers/ir
`imagePullPolicy`	`string`	Image pull policy. One of Always, Never, IfNotPre to Always if :latest tag is specified, or IfNotPrese Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/ir images ↗
`lifecycle`	`object`	Lifecycle is not allowed for ephemeral containers
`livenessProbe`	`object`	Probes are not allowed for ephemeral containers
`name`	`string`	Name of the ephemeral container specified as a This name must be unique among all containers, and ephemeral containers.
`ports`	`array`	Ports are not allowed for ephemeral containers.
`readinessProbe`	`object`	Probes are not allowed for ephemeral containers
`resizePolicy`	`array`	Resources resize policy for the container.

Property	Type	Description
`resources`	`object`	Resources are not allowed for ephemeral contai containers use spare resources already allocate
`restartPolicy`	`string`	Restart policy for the container to manage the re each container within a pod. You cannot set this ephemeral containers.
`restartPolicyRules`	`array`	Represents a list of rules to be checked to deter container should be restarted on exit. You canno ephemeral containers.
`securityContext`	`object`	Optional: SecurityContext defines the security op ephemeral container should be run with. If set, th SecurityContext override the equivalent fields of PodSecurityContext.
`startupProbe`	`object`	Probes are not allowed for ephemeral containers
`stdin`	`boolean`	Whether this container should allocate a buffer fo container runtime. If this is not set, reads from st container will always result in EOF. Default is fals
`stdinOnce`	`boolean`	Whether the container runtime should close the s after it has been opened by a single attach. Whe the stdin stream will remain open across multiple sessions. If stdinOnce is set to true, stdin is oper

Property	Type	Description
		start, is empty until the first client attaches to stdi remains open and accepts data until the client di which time stdin is closed and remains closed ur is restarted. If this flag is false, a container proce from stdin will never receive an EOF. Default is fa
`targetContainerName`	`string`	If set, the name of the container from PodSpec th ephemeral container targets. The ephemeral cor run in the namespaces (IPC, PID, etc) of this cor then the ephemeral container uses the namespa in the Pod spec. The container runtime must implement support fo the runtime does not support namespace targetin result of setting this field is undefined.
`terminationMessagePath`	`string`	Optional: Path at which the file to which the conta termination message will be written is mounted in container's filesystem. Message written is intend final status, such as an assertion failure messag truncated by the node if greater than 4096 bytes. message length across all containers will be limit Defaults to /dev/termination-log. Cannot be upda

Property	Type	Description
`terminationMessagePolicy`	`string`	Indicate how the termination message should be will use the contents of terminationMessagePath container status message on both success and f FallbackToLogsOnError will use the last chunk o output if the termination message file is empty ar exited with an error. The log output is limited to 2 lines, whichever is smaller. Defaults to File. Canr
`tty`	`boolean`	Whether this container should allocate a TTY for requires 'stdin' to be true. Default is false.
`volumeDevices`	`array`	volumeDevices is the list of block devices to be u container.
`volumeMounts`	`array`	Pod volumes to mount into the container's filesys mounts are not allowed for ephemeral containers updated.
`workingDir`	`string`	Container's working directory. If not specified, the runtime's default will be used, which might be co container image. Cannot be updated.

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].args

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].args[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].command

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].command[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].env

Description

List of environment variables to set in the container. Cannot be updated.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].env[]

Description

EnvVar represents an environment variable present in a Container.

Type

`object`

Required

`name`

Property	Type	Description
`name`	`string`	Name of the environment variable. May consist of any printable ASCII characters except '='.

Property	Type	Description
`value`	`string`	Variable references $(VAR_NAME) are expanded using the previously defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input string will be unchanged. Double $$ are reduced to a single $, which allows for escaping the $(VAR_NAME) syntax: i.e. "$$(VAR_NAME)" will produce the string literal "$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".
`valueFrom`	`object`	Source for the environment variable's value. Cannot be used if value is not empty.

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].env[].valueFrom

Description

Source for the environment variable's value. Cannot be used if value is not empty.

Type

`object`

Property	Type	Description
`configMapKeyRef`	`object`	Selects a key of a ConfigMap.
`fieldRef`	`object`	Selects a field of the pod: supports metadata.name, metadata.namespace,

Property	Type	Description
		`metadata.labels['<KEY>']`, `metadata.annotations['<KEY>']`, spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP, status.podIPs.
`fileKeyRef`	`object`	FileKeyRef selects a key of the env file. Requires the EnvFiles feature gate to be enabled.
`resourceFieldRef`	`object`	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.
`secretKeyRef`	`object`	Selects a key of a secret in the pod's namespace

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].env[].valueFrom.configMapKeyRef

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

Property	Type	Description
`key`	`string`	The key to select.
`name`	`string`	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`optional`	`boolean`	Specify whether the ConfigMap or its key must be defined

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].env[].valueFrom.fieldRef

Description

Selects a field of the pod: supports metadata.name, metadata.namespace, `metadata.labels['<KEY>']`, `metadata.annotations['<KEY>']`, spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP, status.podIPs.

Type

`object`

Required

`fieldPath`

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

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].env[].valueFrom.fileKeyRef

Description

FileKeyRef selects a key of the env file. Requires the EnvFiles feature gate to be enabled.

Type

`object`

Required

`key` `path` `volumeName`

Property	Type	Description
`key`	`string`	The key within the env file. An invalid key will prevent the pod from starting. The keys defined within a source may consist of any printable ASCII characters except '='. During Alpha stage of the EnvFiles feature gate, the key size is limited to 128 characters.
`optional`	`boolean`	Specify whether the file or its key must be defined. If the file or key does not exist, then the env var is not published. If optional is set to true and the specified key does not

Property	Type	Description
		exist, the environment variable will not be set in the Pod's containers. If optional is set to false and the specified key does not exist, an error will be returned during Pod creation.
`path`	`string`	The path within the volume from which to select the file. Must be relative and may not contain the '..' path or start with '..'.
`volumeName`	`string`	The name of the volume mount containing the env file.

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].env[].valueFrom.resourceFieldRef

Description

Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.

Type

`object`

Required

`resource`

Property	Type	Description
`containerName`	`string`	Container name: required for volumes, optional for env vars

Property	Type	Description
`divisor`		Specifies the output format of the exposed resources, defaults to "1"
`resource`	`string`	Required: resource to select

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].env[].valueFrom.secretKeyRef

Description

Selects a key of a secret in the pod's namespace

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

Property	Type	Description
`optional`	`boolean`	Specify whether the Secret or its key must be defined

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].envFrom

Description

List of sources to populate environment variables in the container. The keys defined within a source may consist of any printable ASCII characters except '='. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].envFrom[]

Description

EnvFromSource represents the source of a set of ConfigMaps or Secrets

Type

`object`

Property	Type	Description
`configMapRef`	`object`	The ConfigMap to select from

Property	Type	Description
`prefix`	`string`	Optional text to prepend to the name of each environment variable. May consist of any printable ASCII characters except '='.
`secretRef`	`object`	The Secret to select from

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].envFrom[].configMapRef

Description

The ConfigMap to select from

Type

`object`

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

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].envFrom[].secretRef

Description

The Secret to select from

Type

`object`

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

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].lifecycle

Description

Lifecycle is not allowed for ephemeral containers.

Type

`object`

Property	Type	Description
`postStart`	`object`	PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks ↗
`preStop`	`object`	PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks ↗
`stopSignal`	`string`	StopSignal defines which signal will be sent to a container when it is being stopped. If not specified, the default is defined by the container runtime in use. StopSignal can only be set for Pods with a non-empty .spec.os.name

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].lifecycle.postStart

Description

PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks

Type

`object`

Property	Type	Description
`exec`	`object`	Exec specifies a command to execute in the container.
`httpGet`	`object`	HTTPGet specifies an HTTP GET request to perform.
`sleep`	`object`	Sleep represents a duration that the container should sleep.
`tcpSocket`	`object`	Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for backward compatibility. There is no validation of this field and lifecycle hooks will fail at runtime when it is specified.

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].lifecycle.postStart.exec

Description

Exec specifies a command to execute in the container.

Type

`object`

Property	Type	Description
`command`	`array`	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].lifecycle.postStart.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].lifecycle.postStart.exec.command[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].lifecycle.postStart.httpGet

Description

HTTPGet specifies an HTTP GET request to perform.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
`httpHeaders`	`array`	Custom headers to set in the request. HTTP allows repeated headers.
`path`	`string`	Path to access on the HTTP server.
`port`		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
`scheme`	`string`	Scheme to use for connecting to the host. Defaults to HTTP.

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].lifecycle.postStart.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].lifecycle.postStart.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
`value`	`string`	The header field value

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].lifecycle.postStart.sleep

Description

Sleep represents a duration that the container should sleep.

Type

`object`

Required

`seconds`

Property	Type	Description
`seconds`	`integer`	Seconds is the number of seconds to sleep.

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].lifecycle.postStart.tcpSocket

Description

Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for backward compatibility. There is no validation of this field and lifecycle hooks will fail at runtime when it is specified.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
`port`		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].lifecycle.preStop

Description

PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks

Type

`object`

Property	Type	Description
`exec`	`object`	Exec specifies a command to execute in the container.
`httpGet`	`object`	HTTPGet specifies an HTTP GET request to perform.
`sleep`	`object`	Sleep represents a duration that the container should sleep.

Property	Type	Description
`tcpSocket`	`object`	Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for backward compatibility. There is no validation of this field and lifecycle hooks will fail at runtime when it is specified.

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].lifecycle.preStop.exec

Description

Exec specifies a command to execute in the container.

Type

`object`

Property	Type	Description
`command`	`array`	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].lifecycle.preStop.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].lifecycle.preStop.exec.command[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].lifecycle.preStop.httpGet

Description

HTTPGet specifies an HTTP GET request to perform.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.

Property	Type	Description
`httpHeaders`	`array`	Custom headers to set in the request. HTTP allows repeated headers.
`path`	`string`	Path to access on the HTTP server.
`port`		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
`scheme`	`string`	Scheme to use for connecting to the host. Defaults to HTTP.

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].lifecycle.preStop.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].lifecycle.preStop.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
`value`	`string`	The header field value

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].lifecycle.preStop.sleep

Description

Sleep represents a duration that the container should sleep.

Type

`object`

Required

`seconds`

Property	Type	Description
`seconds`	`integer`	Seconds is the number of seconds to sleep.

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].lifecycle.preStop.tcpSocket

Description

Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for backward compatibility. There is no validation of this field and lifecycle hooks will fail at runtime when it is specified.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Optional: Host name to connect to, defaults to the pod IP.
`port`		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].livenessProbe

Description

Probes are not allowed for ephemeral containers.

Type

`object`

Property	Type	Description
`exec`	`object`	Exec specifies a command to execute in th
`failureThreshold`	`integer`	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
`grpc`	`object`	GRPC specifies a GRPC HealthCheckReq
`httpGet`	`object`	HTTPGet specifies an HTTP GET request
`initialDelaySeconds`	`integer`	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗
`periodSeconds`	`integer`	How often (in seconds) to perform the prob 10 seconds. Minimum value is 1.
`successThreshold`	`integer`	Minimum consecutive successes for the pr considered successful after having failed. I Must be 1 for liveness and startup. Minimu
`tcpSocket`	`object`	TCPSocket specifies a connection to a TC

Property	Type	Description
`terminationGracePeriodSeconds`	`integer`	Optional duration in seconds the pod need gracefully upon probe failure. The grace pe duration in seconds after the processes ru pod are sent a termination signal and the ti processes are forcibly halted with a kill sig value longer than the expected cleanup tim process. If this value is nil, the pod's terminationGracePeriodSeconds will be us Otherwise, this value overrides the value p pod spec. Value must be non-negative inte value zero indicates stop immediately via t (no opportunity to shut down). This is a bet requires enabling ProbeTerminationGraceP gate. Minimum value is 1. spec.terminationGracePeriodSeconds is u
`timeoutSeconds`	`integer`	Number of seconds after which the probe t Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].livenessProbe.exec

Description

Exec specifies a command to execute in the container.

Type

`object`

Property	Type	Description
`command`	`array`	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].livenessProbe.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].livenessProbe.exec.command[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].livenessProbe.grpc

Description

GRPC specifies a GRPC HealthCheckRequest.

Type

`object`

Required

`port`

Property	Type	Description
`port`	`integer`	Port number of the gRPC service. Number must be in the range 1 to 65535.
`service`	`string`	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md↗). If this is not specified, the default behavior is defined by gRPC.

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].livenessProbe.httpGet

Description

HTTPGet specifies an HTTP GET request to perform.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
`httpHeaders`	`array`	Custom headers to set in the request. HTTP allows repeated headers.
`path`	`string`	Path to access on the HTTP server.
`port`		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
`scheme`	`string`	Scheme to use for connecting to the host. Defaults to HTTP.

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].livenessProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].livenessProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
`value`	`string`	The header field value

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].livenessProbe.tcpSocket

Description

TCPSocket specifies a connection to a TCP port.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Optional: Host name to connect to, defaults to the pod IP.
`port`		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].ports

Description

Ports are not allowed for ephemeral containers.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].ports[]

Description

ContainerPort represents a network port in a single container.

Type

`object`

Required

`containerPort`

Property	Type	Description
`containerPort`	`integer`	Number of port to expose on the pod's IP address. This must be a valid port number, 0 < x < 65536.
`hostIP`	`string`	What host IP to bind the external port to.
`hostPort`	`integer`	Number of port to expose on the host. If specified, this must be a valid port number, 0 < x < 65536. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this.
`name`	`string`	If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services.
`protocol`	`string`	Protocol for port. Must be UDP, TCP, or SCTP. Defaults to "TCP".

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].readinessProbe

Description

Probes are not allowed for ephemeral containers.

Type

`object`

Property	Type	Description
`exec`	`object`	Exec specifies a command to execute in th
`failureThreshold`	`integer`	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
`grpc`	`object`	GRPC specifies a GRPC HealthCheckReq
`httpGet`	`object`	HTTPGet specifies an HTTP GET request
`initialDelaySeconds`	`integer`	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗
`periodSeconds`	`integer`	How often (in seconds) to perform the prob 10 seconds. Minimum value is 1.
`successThreshold`	`integer`	Minimum consecutive successes for the pr considered successful after having failed. I Must be 1 for liveness and startup. Minimu
`tcpSocket`	`object`	TCPSocket specifies a connection to a TC

Property	Type	Description
`terminationGracePeriodSeconds`	`integer`	Optional duration in seconds the pod need gracefully upon probe failure. The grace pe duration in seconds after the processes ru pod are sent a termination signal and the ti processes are forcibly halted with a kill sig value longer than the expected cleanup tim process. If this value is nil, the pod's terminationGracePeriodSeconds will be us Otherwise, this value overrides the value p pod spec. Value must be non-negative inte value zero indicates stop immediately via t (no opportunity to shut down). This is a bet requires enabling ProbeTerminationGraceP gate. Minimum value is 1. spec.terminationGracePeriodSeconds is u
`timeoutSeconds`	`integer`	Number of seconds after which the probe t Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].readinessProbe.exec

Description

Exec specifies a command to execute in the container.

Type

`object`

Property	Type	Description
`command`	`array`	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].readinessProbe.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].readinessProbe.exec.command[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].readinessProbe.grpc

Description

GRPC specifies a GRPC HealthCheckRequest.

Type

`object`

Required

`port`

Property	Type	Description
`port`	`integer`	Port number of the gRPC service. Number must be in the range 1 to 65535.
`service`	`string`	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md↗). If this is not specified, the default behavior is defined by gRPC.

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].readinessProbe.httpGet

Description

HTTPGet specifies an HTTP GET request to perform.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
`httpHeaders`	`array`	Custom headers to set in the request. HTTP allows repeated headers.
`path`	`string`	Path to access on the HTTP server.
`port`		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
`scheme`	`string`	Scheme to use for connecting to the host. Defaults to HTTP.

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].readinessProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].readinessProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
`value`	`string`	The header field value

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].readinessProbe.tcpSocket

Description

TCPSocket specifies a connection to a TCP port.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Optional: Host name to connect to, defaults to the pod IP.
`port`		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].resizePolicy

Description

Resources resize policy for the container.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].resizePolicy[]

Description

ContainerResizePolicy represents resource resize policy for the container.

Type

`object`

Required

`resourceName` `restartPolicy`

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

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].resources

Description

Resources are not allowed for ephemeral containers. Ephemeral containers use spare resources already allocated to the pod.

Type

`object`

Property	Type	Description
`claims`	`array`	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This field depends on the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
`limits`	`object`	Limits describes the maximum amount of compute resources allowed. More info:

Property	Type	Description
		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.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].resources.claims

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].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.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].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.kubernetesResource.spec.template.spec.ephemeralContainers[].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.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].restartPolicyRules

Description

Represents a list of rules to be checked to determine if the container should be restarted on exit. You cannot set this field on ephemeral containers.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].restartPolicyRules[]

Description

ContainerRestartRule describes how a container exit is handled.

Type

`object`

Required

`action`

Property	Type	Description
`action`	`string`	Specifies the action taken on a container exit if the requirements are satisfied. The only possible value is "Restart" to restart the container.
`exitCodes`	`object`	Represents the exit codes to check on container exits.

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].restartPolicyRules[].exitCodes

Description

Represents the exit codes to check on container exits.

Type

`object`

Required

`operator`

Property	Type	Description
`operator`	`string`	Represents the relationship between the container exit code(s) and the specified values. Possible values are: • In: the requirement is satisfied if the container exit code is in the set of specified values. • NotIn: the requirement is satisfied if the container exit code is not in the set of specified values.
`values`	`array`	Specifies the set of values to check for container exit codes. At most 255 elements are allowed.

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].restartPolicyRules[].exitCodes.values

Description

Specifies the set of values to check for container exit codes. At most 255 elements are allowed.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].restartPolicyRules[].exitCodes.values[]

Type

`integer`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].securityContext

Description

Optional: SecurityContext defines the security options the ephemeral container should be run with. If set, the fields of SecurityContext override the equivalent fields of PodSecurityContext.

Type

`object`

Property	Type	Description
`allowPrivilegeEscalation`	`boolean`	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the no_new_privs flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: 1. run as Privileged 2. has CAP_SYS_ADMIN Note that this field cannot be set when spec.os.name is windows.
`appArmorProfile`	`object`	appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when spec.os.name is windows.
`capabilities`	`object`	The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when spec.os.name is windows.
`privileged`	`boolean`	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
`procMount`	`string`	procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows.
`readOnlyRootFilesystem`	`boolean`	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
`runAsGroup`	`integer`	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
`runAsNonRoot`	`boolean`	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If

Property	Type	Description
		set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
`runAsUser`	`integer`	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
`seLinuxOptions`	`object`	The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
`seccompProfile`	`object`	The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
`windowsOptions`	`object`	The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is linux.

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].securityContext.appArmorProfile

Description

appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Required

`type`

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

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.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].securityContext.capabilities

Description

The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Property	Type	Description
`add`	`array`	Added capabilities
`drop`	`array`	Removed capabilities

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].securityContext.capabilities.add

Description

Added capabilities

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].securityContext.capabilities.add[]

Description

Capability represent POSIX capabilities type

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].securityContext.capabilities.drop

Description

Removed capabilities

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].securityContext.capabilities.drop[]

Description

Capability represent POSIX capabilities type

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].securityContext.seLinuxOptions

Description

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

Type

`object`

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

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].securityContext.seccompProfile

Description

The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Required

`type`

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

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].securityContext.windowsOptions

Description

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

Type

`object`

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

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].startupProbe

Description

Probes are not allowed for ephemeral containers.

Type

`object`

Property	Type	Description
`exec`	`object`	Exec specifies a command to execute in th
`failureThreshold`	`integer`	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
`grpc`	`object`	GRPC specifies a GRPC HealthCheckReq
`httpGet`	`object`	HTTPGet specifies an HTTP GET request
`initialDelaySeconds`	`integer`	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗
`periodSeconds`	`integer`	How often (in seconds) to perform the prob 10 seconds. Minimum value is 1.

Property	Type	Description
`successThreshold`	`integer`	Minimum consecutive successes for the pr considered successful after having failed. I Must be 1 for liveness and startup. Minimu
`tcpSocket`	`object`	TCPSocket specifies a connection to a TC
`terminationGracePeriodSeconds`	`integer`	Optional duration in seconds the pod need gracefully upon probe failure. The grace pe duration in seconds after the processes ru pod are sent a termination signal and the ti processes are forcibly halted with a kill sig value longer than the expected cleanup tim process. If this value is nil, the pod's terminationGracePeriodSeconds will be us Otherwise, this value overrides the value p pod spec. Value must be non-negative inte value zero indicates stop immediately via t (no opportunity to shut down). This is a bet requires enabling ProbeTerminationGraceP gate. Minimum value is 1. spec.terminationGracePeriodSeconds is u
`timeoutSeconds`	`integer`	Number of seconds after which the probe t Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].startupProbe.exec

Description

Exec specifies a command to execute in the container.

Type

`object`

Property	Type	Description
`command`	`array`	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].startupProbe.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].startupProbe.exec.command[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].startupProbe.grpc

Description

GRPC specifies a GRPC HealthCheckRequest.

Type

`object`

Required

`port`

Property	Type	Description
`port`	`integer`	Port number of the gRPC service. Number must be in the range 1 to 65535.
`service`	`string`	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md↗). If this is not specified, the default behavior is defined by gRPC.

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].startupProbe.httpGet

Description

HTTPGet specifies an HTTP GET request to perform.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
`httpHeaders`	`array`	Custom headers to set in the request. HTTP allows repeated headers.
`path`	`string`	Path to access on the HTTP server.
`port`		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
`scheme`	`string`	Scheme to use for connecting to the host. Defaults to HTTP.

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].startupProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].startupProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
`value`	`string`	The header field value

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].startupProbe.tcpSocket

Description

TCPSocket specifies a connection to a TCP port.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Optional: Host name to connect to, defaults to the pod IP.
`port`		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].volumeDevices

Description

volumeDevices is the list of block devices to be used by the container.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].volumeDevices[]

Description

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

Type

`object`

Required

`devicePath` `name`

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

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].volumeMounts

Description

Pod volumes to mount into the container's filesystem. Subpath mounts are not allowed for ephemeral containers. Cannot be updated.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.ephemeralContainers[].volumeMounts[]

Description

VolumeMount describes a mounting of a Volume within a container.

Type

`object`

Required

`mountPath` `name`

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

Property	Type	Description
		field is set to IfPossible, the mount is made recursively read-only, if it is supported by the container runtime. If this field is set to Enabled, the mount is made recursively read-only if it is supported by the container runtime, otherwise the pod will not be started and an error will be generated to indicate the reason. If this field is set to IfPossible or Enabled, MountPropagation must be set to None (or be unspecified, which defaults to None). If this field is not specified, it is treated as an equivalent of Disabled.
`subPath`	`string`	Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).
`subPathExpr`	`string`	Expanded path within the volume from which the container's volume should be mounted. Behaves similarly to SubPath but environment variable references $(VAR_NAME) are expanded using the container's environment. Defaults to "" (volume's root). SubPathExpr and SubPath are mutually exclusive.

.spec.resources.kubernetesResource.spec.template.spec.hostAliases

Description

HostAliases is an optional list of hosts and IPs that will be injected into the pod's hosts file if specified.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.hostAliases[]

Description

HostAlias holds the mapping between IP and hostnames that will be injected as an entry in the pod's hosts file.

Type

`object`

Required

`ip`

Property	Type	Description
`hostnames`	`array`	Hostnames for the above IP address.
`ip`	`string`	IP address of the host file entry.

.spec.resources.kubernetesResource.spec.template.spec.hostAliases[].hostnames

Description

Hostnames for the above IP address.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.hostAliases[].hostnames[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.imagePullSecrets

Description

ImagePullSecrets is an optional list of references to secrets in the same namespace to use for pulling any of the images used by this PodSpec. If specified, these secrets will be passed to individual puller implementations for them to use. More info: https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.imagePullSecrets[]

Description

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

Type

`object`

Property	Type	Description
`name`	`string`	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost

Property	Type	Description
		certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

.spec.resources.kubernetesResource.spec.template.spec.initContainers

Description

List of initialization containers belonging to the pod. Init containers are executed in order prior to containers being started. If any init container fails, the pod is considered to have failed and is handled according to its restartPolicy. The name for an init container or normal container must be unique among all containers. Init containers may not have Lifecycle actions, Readiness probes, Liveness probes, or Startup probes. The resourceRequirements of an init container are taken into account during scheduling by finding the highest request/limit for each resource type, and then using the max of that value or the sum of the normal containers. Limits are applied to init containers in a similar fashion. Init containers cannot currently be added or removed. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/init-containers/

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[]

Description

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

Type

`object`

Required

`name`

Property	Type	Description
`args`	`array`	Arguments to the entrypoint. The container imag if this is not provided. Variable references $(VAR expanded using the container's environment. If a be resolved, the reference in the input string will Double $$ are reduced to a single $, which allow the $(VAR_NAME) syntax: i.e. "$$(VAR_NAME)' string literal "$(VAR_NAME)". Escaped reference expanded, regardless of whether the variable ex Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-appli command-argument-container/#running-a-comm
`command`	`array`	Entrypoint array. Not executed within a shell. The image's ENTRYPOINT is used if this is not provi references $(VAR_NAME) are expanded using tl environment. If a variable cannot be resolved, th the input string will be unchanged. Double $$ are single $, which allows for escaping the $(VAR_N i.e. "$$(VAR_NAME)" will produce the string liter "$(VAR_NAME)". Escaped references will never regardless of whether the variable exists or not. updated. More info: https://kubernetes.io/docs/ta application/define-command-argument-container command-in-a-shell ↗
`env`	`array`	List of environment variables to set in the contair updated.
`envFrom`	`array`	List of sources to populate environment variables container. The keys defined within a source may

Property	Type	Description
		printable ASCII characters except '='. When a ke multiple sources, the value associated with the la take precedence. Values defined by an Env with will take precedence. Cannot be updated.
`image`	`string`	Container image name. More info: https://kubernetes.io/docs/concepts/containers/ir field is optional to allow higher level config mana default or override container images in workload Deployments and StatefulSets.
`imagePullPolicy`	`string`	Image pull policy. One of Always, Never, IfNotPre to Always if :latest tag is specified, or IfNotPrese Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/ir images ↗
`lifecycle`	`object`	Actions that the management system should tak container lifecycle events. Cannot be updated.
`livenessProbe`	`object`	Periodic probe of container liveness. Container w the probe fails. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/p lifecycle#container-probes ↗

Property	Type	Description
`name`	`string`	Name of the container specified as a DNS_LABE container in a pod must have a unique name (DN Cannot be updated.
`ports`	`array`	List of ports to expose from the container. Not sp here DOES NOT prevent that port from being ex which is listening on the default "0.0.0.0" addres container will be accessible from the network. Mo array with strategic merge patch may corrupt the information See https://github.com/kubernetes/kubernetes/issues Cannot be updated.
`readinessProbe`	`object`	Periodic probe of container service readiness. Co removed from service endpoints if the probe fails updated. More info: https://kubernetes.io/docs/concepts/workloads/po lifecycle#container-probes ↗
`resizePolicy`	`array`	Resources resize policy for the container. This fie set on ephemeral containers.
`resources`	`object`	Compute Resources required by this container. C updated. More info: https://kubernetes.io/docs/concepts/configuration resources-containers/ ↗

Property	Type	Description
`restartPolicy`	`string`	RestartPolicy defines the restart behavior of indiv in a pod. This overrides the pod-level restart poli field is not specified, the restart behavior is defin restart policy and the container type. Additionally RestartPolicy as "Always" for the init container w following effect: this init container will be continua exit until all regular containers have terminated. (containers have completed, all init containers wit "Always" will be shut down. This lifecycle differs f containers and is often referred to as a "sidecar" Although this init container still starts in the init co sequence, it does not wait for the container to co proceeding to the next init container. Instead, the container starts immediately after this init contair after any startupProbe has successfully complete
`restartPolicyRules`	`array`	Represents a list of rules to be checked to deterr container should be restarted on exit. The rules a order. Once a rule matches a container exit cond remaining rules are ignored. If no rule matches th condition, the Container-level restart policy deter whether the container is restarted or not. Constra rules: • At most 20 rules are allowed. • Rules can have the same action. • Identical rules are not forbidden in validations are specified, container MUST set RestartPol even it if matches the Pod's RestartPolicy.

Property	Type	Description
`securityContext`	`object`	SecurityContext defines the security options the be run with. If set, the fields of SecurityContext o equivalent fields of PodSecurityContext. More inf https://kubernetes.io/docs/tasks/configure-pod-container/security-context/ ↗
`startupProbe`	`object`	StartupProbe indicates that the Pod has success If specified, no other probes are executed until th successfully. If this probe fails, the Pod will be re the livenessProbe failed. This can be used to pro probe parameters at the beginning of a Pod's life might take a long time to load data or warm a ca steady-state operation. This cannot be updated. https://kubernetes.io/docs/concepts/workloads/p lifecycle#container-probes ↗
`stdin`	`boolean`	Whether this container should allocate a buffer fo container runtime. If this is not set, reads from st container will always result in EOF. Default is fals
`stdinOnce`	`boolean`	Whether the container runtime should close the s after it has been opened by a single attach. Whe the stdin stream will remain open across multiple sessions. If stdinOnce is set to true, stdin is oper start, is empty until the first client attaches to stdi remains open and accepts data until the client di which time stdin is closed and remains closed ur is restarted. If this flag is false, a container proce from stdin will never receive an EOF. Default is fa

Property	Type	Description
`terminationMessagePath`	`string`	Optional: Path at which the file to which the conta termination message will be written is mounted in container's filesystem. Message written is intend final status, such as an assertion failure messag truncated by the node if greater than 4096 bytes. message length across all containers will be limit Defaults to /dev/termination-log. Cannot be upda
`terminationMessagePolicy`	`string`	Indicate how the termination message should be will use the contents of terminationMessagePath container status message on both success and f FallbackToLogsOnError will use the last chunk of output if the termination message file is empty an exited with an error. The log output is limited to 2 lines, whichever is smaller. Defaults to File. Cann
`tty`	`boolean`	Whether this container should allocate a TTY for requires 'stdin' to be true. Default is false.
`volumeDevices`	`array`	volumeDevices is the list of block devices to be u container.
`volumeMounts`	`array`	Pod volumes to mount into the container's filesys updated.
`workingDir`	`string`	Container's working directory. If not specified, the runtime's default will be used, which might be co

Property	Type	Description
		container image. Cannot be updated.

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].args

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].args[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].command

Description

Entrypoint array. Not executed within a shell. The container image's ENTRYPOINT is used if this is not provided. Variable references $(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string

will be unchanged. Double $$ are reduced to a single $, which allows for escaping the $(VAR_NAME) syntax: i.e. "$$(VAR_NAME)" will produce the string literal "$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].command[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].env

Description

List of environment variables to set in the container. Cannot be updated.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].env[]

Description

EnvVar represents an environment variable present in a Container.

Type

`object`

Required

`name`

Property	Type	Description
`name`	`string`	Name of the environment variable. May consist of any printable ASCII characters except '='.
`value`	`string`	Variable references $(VAR_NAME) are expanded using the previously defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input string will be unchanged. Double $$ are reduced to a single $, which allows for escaping the $(VAR_NAME) syntax: i.e. "$$(VAR_NAME)" will produce the string literal "$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".
`valueFrom`	`object`	Source for the environment variable's value. Cannot be used if value is not empty.

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].env[].valueFrom

Description

Source for the environment variable's value. Cannot be used if value is not empty.

Type

`object`

Property	Type	Description
`configMapKeyRef`	`object`	Selects a key of a ConfigMap.
`fieldRef`	`object`	Selects a field of the pod: supports metadata.name, metadata.namespace, `metadata.labels['<KEY>']`, `metadata.annotations['<KEY>']`, spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP, status.podIPs.
`fileKeyRef`	`object`	FileKeyRef selects a key of the env file. Requires the EnvFiles feature gate to be enabled.
`resourceFieldRef`	`object`	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.
`secretKeyRef`	`object`	Selects a key of a secret in the pod's namespace

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].env[].valueFrom.configMapKeyRef

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

Property	Type	Description
`key`	`string`	The key to select.
`name`	`string`	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`optional`	`boolean`	Specify whether the ConfigMap or its key must be defined

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].env[].valueFrom.fieldRef

Description

Selects a field of the pod: supports metadata.name, metadata.namespace, `metadata.labels['<KEY>']`, `metadata.annotations['<KEY>']`, spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP, status.podIPs.

Type

`object`

Required

`fieldPath`

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

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].env[].valueFrom.fileKeyRef

Description

FileKeyRef selects a key of the env file. Requires the EnvFiles feature gate to be enabled.

Type

`object`

Required

`key` `path` `volumeName`

Property	Type	Description
`key`	`string`	The key within the env file. An invalid key will prevent the pod from starting. The keys defined within a source may consist of any printable ASCII characters except '='. During Alpha stage of the EnvFiles feature gate, the key size is limited to 128 characters.
`optional`	`boolean`	Specify whether the file or its key must be defined. If the file or key does not exist, then the env var is not published. If optional is set to true and the specified key does not

Property	Type	Description
		exist, the environment variable will not be set in the Pod's containers. If optional is set to false and the specified key does not exist, an error will be returned during Pod creation.
`path`	`string`	The path within the volume from which to select the file. Must be relative and may not contain the '..' path or start with '..'.
`volumeName`	`string`	The name of the volume mount containing the env file.

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].env[].valueFrom.resourceFieldRef

Description

Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.

Type

`object`

Required

`resource`

Property	Type	Description
`containerName`	`string`	Container name: required for volumes, optional for env vars

Property	Type	Description
`divisor`		Specifies the output format of the exposed resources, defaults to "1"
`resource`	`string`	Required: resource to select

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].env[].valueFrom.secretKeyRef

Description

Selects a key of a secret in the pod's namespace

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

Property	Type	Description
`optional`	`boolean`	Specify whether the Secret or its key must be defined

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].envFrom

Description

List of sources to populate environment variables in the container. The keys defined within a source may consist of any printable ASCII characters except '='. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].envFrom[]

Description

EnvFromSource represents the source of a set of ConfigMaps or Secrets

Type

`object`

Property	Type	Description
`configMapRef`	`object`	The ConfigMap to select from

Property	Type	Description
`prefix`	`string`	Optional text to prepend to the name of each environment variable. May consist of any printable ASCII characters except '='.
`secretRef`	`object`	The Secret to select from

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].envFrom[].configMapRef

Description

The ConfigMap to select from

Type

`object`

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

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].envFrom[].secretRef

Description

The Secret to select from

Type

`object`

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

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].lifecycle

Description

Actions that the management system should take in response to container lifecycle events. Cannot be updated.

Type

`object`

Property	Type	Description
`postStart`	`object`	PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks ↗
`preStop`	`object`	PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks ↗
`stopSignal`	`string`	StopSignal defines which signal will be sent to a container when it is being stopped. If not specified, the default is defined by the container runtime in use. StopSignal can only be set for Pods with a non-empty .spec.os.name

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].lifecycle.postStart

Description

PostStart is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks

Type

`object`

Property	Type	Description
`exec`	`object`	Exec specifies a command to execute in the container.
`httpGet`	`object`	HTTPGet specifies an HTTP GET request to perform.
`sleep`	`object`	Sleep represents a duration that the container should sleep.
`tcpSocket`	`object`	Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for backward compatibility. There is no validation of this field and lifecycle hooks will fail at runtime when it is specified.

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].lifecycle.postStart.exec

Description

Exec specifies a command to execute in the container.

Type

`object`

Property	Type	Description
`command`	`array`	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].lifecycle.postStart.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].lifecycle.postStart.exec.command[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].lifecycle.postStart.httpGet

Description

HTTPGet specifies an HTTP GET request to perform.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
`httpHeaders`	`array`	Custom headers to set in the request. HTTP allows repeated headers.
`path`	`string`	Path to access on the HTTP server.
`port`		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
`scheme`	`string`	Scheme to use for connecting to the host. Defaults to HTTP.

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].lifecycle.postStart.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].lifecycle.postStart.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
`value`	`string`	The header field value

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].lifecycle.postStart.sleep

Description

Sleep represents a duration that the container should sleep.

Type

`object`

Required

`seconds`

Property	Type	Description
`seconds`	`integer`	Seconds is the number of seconds to sleep.

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].lifecycle.postStart.tcpSocket

Description

Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for backward compatibility. There is no validation of this field and lifecycle hooks will fail at runtime when it is specified.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Optional: Host name to connect to, defaults to the pod IP.
`port`		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].lifecycle.preStop

Description

PreStop is called immediately before a container is terminated due to an API request or management event such as liveness/startup probe failure, preemption, resource contention, etc. The handler is not called if the container crashes or exits. The Pod's termination grace period countdown begins before the PreStop hook is executed. Regardless of the outcome of the handler, the container will eventually terminate within the Pod's termination grace period (unless delayed by finalizers). Other management of the container blocks until the hook completes or until the termination grace period is reached. More info: https://kubernetes.io/docs/concepts/containers/container-lifecycle-hooks/#container-hooks

Type

`object`

Property	Type	Description
`exec`	`object`	Exec specifies a command to execute in the container.
`httpGet`	`object`	HTTPGet specifies an HTTP GET request to perform.
`sleep`	`object`	Sleep represents a duration that the container should sleep.
`tcpSocket`	`object`	Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for backward compatibility. There is no validation of this field and lifecycle hooks will fail at runtime when it is specified.

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].lifecycle.preStop.exec

Description

Exec specifies a command to execute in the container.

Type

`object`

Property	Type	Description
`command`	`array`	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].lifecycle.preStop.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].lifecycle.preStop.exec.command[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].lifecycle.preStop.httpGet

Description

HTTPGet specifies an HTTP GET request to perform.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
`httpHeaders`	`array`	Custom headers to set in the request. HTTP allows repeated headers.
`path`	`string`	Path to access on the HTTP server.

Property	Type	Description
`port`		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
`scheme`	`string`	Scheme to use for connecting to the host. Defaults to HTTP.

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
`value`	`string`	The header field value

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].lifecycle.preStop.sleep

Description

Sleep represents a duration that the container should sleep.

Type

`object`

Required

`seconds`

Property	Type	Description
`seconds`	`integer`	Seconds is the number of seconds to sleep.

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].lifecycle.preStop.tcpSocket

Description

Deprecated. TCPSocket is NOT supported as a LifecycleHandler and kept for backward compatibility. There is no validation of this field and lifecycle hooks will fail at runtime when it

is specified.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Optional: Host name to connect to, defaults to the pod IP.
`port`		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].livenessProbe

Description

Periodic probe of container liveness. Container will be restarted if the probe fails. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

Type

`object`

Property	Type	Description
`exec`	`object`	Exec specifies a command to execute in th

Property	Type	Description
`failureThreshold`	`integer`	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
`grpc`	`object`	GRPC specifies a GRPC HealthCheckReq
`httpGet`	`object`	HTTPGet specifies an HTTP GET request
`initialDelaySeconds`	`integer`	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗
`periodSeconds`	`integer`	How often (in seconds) to perform the prob 10 seconds. Minimum value is 1.
`successThreshold`	`integer`	Minimum consecutive successes for the pr considered successful after having failed. I Must be 1 for liveness and startup. Minimu
`tcpSocket`	`object`	TCPSocket specifies a connection to a TC
`terminationGracePeriodSeconds`	`integer`	Optional duration in seconds the pod need gracefully upon probe failure. The grace pe duration in seconds after the processes ru

Property	Type	Description
		pod are sent a termination signal and the ti processes are forcibly halted with a kill sigı value longer than the expected cleanup tim process. If this value is nil, the pod's terminationGracePeriodSeconds will be us Otherwise, this value overrides the value p pod spec. Value must be non-negative inte value zero indicates stop immediately via t (no opportunity to shut down). This is a bet requires enabling ProbeTerminationGraceF gate. Minimum value is 1. spec.terminationGracePeriodSeconds is u
`timeoutSeconds`	`integer`	Number of seconds after which the probe t Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].livenessProbe.exec

Description

Exec specifies a command to execute in the container.

Type

`object`

Property	Type	Description
`command`	`array`	Command is the command line to execute inside the container, the working directory for the command is root

Property	Type	Description
		('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].livenessProbe.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].livenessProbe.exec.command[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].livenessProbe.grpc

Description

GRPC specifies a GRPC HealthCheckRequest.

Type

`object`

Required

`port`

Property	Type	Description
`port`	`integer`	Port number of the gRPC service. Number must be in the range 1 to 65535.
`service`	`string`	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md↗). If this is not specified, the default behavior is defined by gRPC.

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].livenessProbe.httpGet

Description

HTTPGet specifies an HTTP GET request to perform.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
`httpHeaders`	`array`	Custom headers to set in the request. HTTP allows repeated headers.
`path`	`string`	Path to access on the HTTP server.
`port`		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
`scheme`	`string`	Scheme to use for connecting to the host. Defaults to HTTP.

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].livenessProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].livenessProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
`value`	`string`	The header field value

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].livenessProbe.tcpSocket

Description

TCPSocket specifies a connection to a TCP port.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Optional: Host name to connect to, defaults to the pod IP.

Property	Type	Description
`port`		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].ports

Description

List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address inside a container will be accessible from the network. Modifying this array with strategic merge patch may corrupt the data. For more information See https://github.com/kubernetes/kubernetes/issues/108255. Cannot be updated.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].ports[]

Description

ContainerPort represents a network port in a single container.

Type

`object`

Required

`containerPort`

Property	Type	Description
`containerPort`	`integer`	Number of port to expose on the pod's IP address. This must be a valid port number, 0 < x < 65536.
`hostIP`	`string`	What host IP to bind the external port to.
`hostPort`	`integer`	Number of port to expose on the host. If specified, this must be a valid port number, 0 < x < 65536. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this.
`name`	`string`	If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services.
`protocol`	`string`	Protocol for port. Must be UDP, TCP, or SCTP. Defaults to "TCP".

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].readinessProbe

Description

Periodic probe of container service readiness. Container will be removed from service endpoints if the probe fails. Cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

Type

`object`

Property	Type	Description
`exec`	`object`	Exec specifies a command to execute in th
`failureThreshold`	`integer`	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
`grpc`	`object`	GRPC specifies a GRPC HealthCheckReq
`httpGet`	`object`	HTTPGet specifies an HTTP GET request
`initialDelaySeconds`	`integer`	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗
`periodSeconds`	`integer`	How often (in seconds) to perform the prob 10 seconds. Minimum value is 1.
`successThreshold`	`integer`	Minimum consecutive successes for the pr considered successful after having failed. I Must be 1 for liveness and startup. Minimu

Property	Type	Description
`tcpSocket`	`object`	TCPSocket specifies a connection to a TC
`terminationGracePeriodSeconds`	`integer`	Optional duration in seconds the pod need gracefully upon probe failure. The grace pe duration in seconds after the processes ru pod are sent a termination signal and the ti processes are forcibly halted with a kill sig value longer than the expected cleanup tim process. If this value is nil, the pod's terminationGracePeriodSeconds will be us Otherwise, this value overrides the value p pod spec. Value must be non-negative inte value zero indicates stop immediately via t (no opportunity to shut down). This is a bet requires enabling ProbeTerminationGraceP gate. Minimum value is 1. spec.terminationGracePeriodSeconds is u
`timeoutSeconds`	`integer`	Number of seconds after which the probe t Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].readinessProbe.exec

Description

Exec specifies a command to execute in the container.

Type

`object`

Property	Type	Description
`command`	`array`	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].readinessProbe.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].readinessProbe.exec.command[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].readinessProbe.grpc

Description

GRPC specifies a GRPC HealthCheckRequest.

Type

`object`

Required

`port`

Property	Type	Description
`port`	`integer`	Port number of the gRPC service. Number must be in the range 1 to 65535.
`service`	`string`	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md↗). If this is not specified, the default behavior is defined by gRPC.

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].readinessProbe.httpGet

Description

HTTPGet specifies an HTTP GET request to perform.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
`httpHeaders`	`array`	Custom headers to set in the request. HTTP allows repeated headers.
`path`	`string`	Path to access on the HTTP server.
`port`		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
`scheme`	`string`	Scheme to use for connecting to the host. Defaults to HTTP.

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].readinessProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].readinessProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
`value`	`string`	The header field value

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].readinessProbe.tcpSocket

Description

TCPSocket specifies a connection to a TCP port.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Optional: Host name to connect to, defaults to the pod IP.
`port`		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].resizePolicy

Description

Resources resize policy for the container. This field cannot be set on ephemeral containers.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].resizePolicy[]

Description

ContainerResizePolicy represents resource resize policy for the container.

Type

`object`

Required

`resourceName` `restartPolicy`

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

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].resources

Description

Compute Resources required by this container. Cannot be updated. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

Type

`object`

Property	Type	Description
`claims`	`array`	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This field depends on the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
`limits`	`object`	Limits describes the maximum amount of compute resources allowed. More info:

Property	Type	Description
		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.resources.kubernetesResource.spec.template.spec.initContainers[].resources.claims

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].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.resources.kubernetesResource.spec.template.spec.initContainers[].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.kubernetesResource.spec.template.spec.initContainers[].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.resources.kubernetesResource.spec.template.spec.initContainers[].restartPolicyRules

Description

Represents a list of rules to be checked to determine if the container should be restarted on exit. The rules are evaluated in order. Once a rule matches a container exit condition, the remaining rules are ignored. If no rule matches the container exit condition, the Container-level restart policy determines the whether the container is restarted or not. Constraints on the rules: - At most 20 rules are allowed. - Rules can have the same action. - Identical rules are not forbidden in validations. When rules are specified, container MUST set RestartPolicy explicitly even it if matches the Pod's RestartPolicy.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].restartPolicyRules[]

Description

ContainerRestartRule describes how a container exit is handled.

Type

`object`

Required

`action`

Property	Type	Description
`action`	`string`	Specifies the action taken on a container exit if the requirements are satisfied. The only possible value is "Restart" to restart the container.

Property	Type	Description
`exitCodes`	`object`	Represents the exit codes to check on container exits.

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].restartPolicyRules[].exitCodes

Description

Represents the exit codes to check on container exits.

Type

`object`

Required

`operator`

Property	Type	Description
`operator`	`string`	Represents the relationship between the container exit code(s) and the specified values. Possible values are: • In: the requirement is satisfied if the container exit code is in the set of specified values. • NotIn: the requirement is satisfied if the container exit code is not in the set of specified values.
`values`	`array`	Specifies the set of values to check for container exit codes. At most 255 elements are allowed.

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].restartPolicyRules[].exitCodes.values

Description

Specifies the set of values to check for container exit codes. At most 255 elements are allowed.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].restartPolicyRules[].exitCodes.values[]

Type

`integer`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].securityContext

Description

SecurityContext defines the security options the container should be run with. If set, the fields of SecurityContext override the equivalent fields of PodSecurityContext. More info: https://kubernetes.io/docs/tasks/configure-pod-container/security-context/

Type

`object`

Property	Type	Description
`allowPrivilegeEscalation`	`boolean`	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the no_new_privs flag will be set on the

Property	Type	Description
		container process. AllowPrivilegeEscalation is true always when the container is: 1. run as Privileged 2. has CAP_SYS_ADMIN Note that this field cannot be set when spec.os.name is windows.
`appArmorProfile`	`object`	appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when spec.os.name is windows.
`capabilities`	`object`	The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when spec.os.name is windows.
`privileged`	`boolean`	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.
`procMount`	`string`	procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked

Property	Type	Description
		paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows.
`readOnlyRootFilesystem`	`boolean`	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
`runAsGroup`	`integer`	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
`runAsNonRoot`	`boolean`	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

Property	Type	Description
`runAsUser`	`integer`	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
`seLinuxOptions`	`object`	The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
`seccompProfile`	`object`	The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.
`windowsOptions`	`object`	The Windows specific settings applied to all containers. If unspecified, the options from the PodSecurityContext will be used. If set

Property	Type	Description
		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.resources.kubernetesResource.spec.template.spec.initContainers[].securityContext.appArmorProfile

Description

appArmorProfile is the AppArmor options to use by this container. If set, this profile overrides the pod's appArmorProfile. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Required

`type`

Property	Type	Description
`localhostProfile`	`string`	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
`type`	`string`	type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container

Property	Type	Description
		runtime's default profile. Unconfined - no AppArmor enforcement.

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].securityContext.capabilities

Description

The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Property	Type	Description
`add`	`array`	Added capabilities
`drop`	`array`	Removed capabilities

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].securityContext.capabilities.add

Description

Added capabilities

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].securityContext.capabilities.add[]

Description

Capability represent POSIX capabilities type

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].securityContext.capabilities.drop

Description

Removed capabilities

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].securityContext.capabilities.drop[]

Description

Capability represent POSIX capabilities type

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].securityContext.seLinuxOptions

Description

The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value

specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

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

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].securityContext.seccompProfile

Description

The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options. Note that this field cannot be set when spec.os.name is windows.

Type

`object`

Required

`type`

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

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].securityContext.windowsOptions

Description

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

Type

`object`

Property	Type	Description
`gmsaCredentialSpec`	`string`	GMSACredentialSpec is where the GMSA admission webhook

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

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].startupProbe

Description

StartupProbe indicates that the Pod has successfully initialized. If specified, no other probes are executed until this completes successfully. If this probe fails, the Pod will be restarted, just as if the livenessProbe failed. This can be used to provide different probe parameters at the beginning of a Pod's lifecycle, when it might take a long time to load data or warm a cache, than during steady-state operation. This cannot be updated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

Type

`object`

Property	Type	Description
`exec`	`object`	Exec specifies a command to execute in th
`failureThreshold`	`integer`	Minimum consecutive failures for the probe considered failed after having succeeded. Minimum value is 1.
`grpc`	`object`	GRPC specifies a GRPC HealthCheckReq
`httpGet`	`object`	HTTPGet specifies an HTTP GET request
`initialDelaySeconds`	`integer`	Number of seconds after the container has before liveness probes are initiated. More i https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗

Property	Type	Description
`periodSeconds`	`integer`	How often (in seconds) to perform the prob 10 seconds. Minimum value is 1.
`successThreshold`	`integer`	Minimum consecutive successes for the pr considered successful after having failed. I Must be 1 for liveness and startup. Minimu
`tcpSocket`	`object`	TCPSocket specifies a connection to a TC
`terminationGracePeriodSeconds`	`integer`	Optional duration in seconds the pod need gracefully upon probe failure. The grace pe duration in seconds after the processes run pod are sent a termination signal and the ti processes are forcibly halted with a kill sign value longer than the expected cleanup tim process. If this value is nil, the pod's terminationGracePeriodSeconds will be us Otherwise, this value overrides the value p pod spec. Value must be non-negative inte value zero indicates stop immediately via t (no opportunity to shut down). This is a bet requires enabling ProbeTerminationGraceP gate. Minimum value is 1. spec.terminationGracePeriodSeconds is us

Property	Type	Description
`timeoutSeconds`	`integer`	Number of seconds after which the probe t Defaults to 1 second. Minimum value is 1. https://kubernetes.io/docs/concepts/worklo lifecycle#container-probes ↗

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].startupProbe.exec

Description

Exec specifies a command to execute in the container.

Type

`object`

Property	Type	Description
`command`	`array`	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].startupProbe.exec.command

Description

Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions ('|', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].startupProbe.exec.command[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].startupProbe.grpc

Description

GRPC specifies a GRPC HealthCheckRequest.

Type

`object`

Required

`port`

Property	Type	Description
`port`	`integer`	Port number of the gRPC service. Number must be in the range 1 to 65535.
`service`	`string`	Service is the name of the service to place in the gRPC HealthCheckRequest (see

Property	Type	Description
		https://github.com/grpc/grpc/blob/master/doc/health-checking.md ↗). If this is not specified, the default behavior is defined by gRPC.

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].startupProbe.httpGet

Description

HTTPGet specifies an HTTP GET request to perform.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
`httpHeaders`	`array`	Custom headers to set in the request. HTTP allows repeated headers.
`path`	`string`	Path to access on the HTTP server.

Property	Type	Description
`port`		Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
`scheme`	`string`	Scheme to use for connecting to the host. Defaults to HTTP.

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].startupProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].startupProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
`value`	`string`	The header field value

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].startupProbe.tcpSocket

Description

TCPSocket specifies a connection to a TCP port.

Type

`object`

Required

`port`

Property	Type	Description
`host`	`string`	Optional: Host name to connect to, defaults to the pod IP.
`port`		Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].volumeDevices

Description

volumeDevices is the list of block devices to be used by the container.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].volumeDevices[]

Description

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

Type

`object`

Required

`devicePath` `name`

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

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].volumeMounts

Description

Pod volumes to mount into the container's filesystem. Cannot be updated.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.initContainers[].volumeMounts[]

Description

VolumeMount describes a mounting of a Volume within a container.

Type

`object`

Required

`mountPath` `name`

Property	Type	Description
`mountPath`	`string`	Path within the container at which the volume should be mounted. Must not contain ':'.
`mountPropagation`	`string`	mountPropagation determines how mounts are propagated from the host to container and the other way around. When not set, MountPropagationNone is used. This field is beta in 1.10. When RecursiveReadOnly is set to IfPossible or to Enabled, MountPropagation must be None or unspecified (which defaults to None).
`name`	`string`	This must match the Name of a Volume.

Property	Type	Description
`readOnly`	`boolean`	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.
`recursiveReadOnly`	`string`	RecursiveReadOnly specifies whether read-only mounts should be handled recursively. If ReadOnly is false, this field has no meaning and must be unspecified. If ReadOnly is true, and this field is set to Disabled, the mount is not made recursively read-only. If this field is set to IfPossible, the mount is made recursively read-only, if it is supported by the container runtime. If this field is set to Enabled, the mount is made recursively read-only if it is supported by the container runtime, otherwise the pod will not be started and an error will be generated to indicate the reason. If this field is set to IfPossible or Enabled, MountPropagation must be set to None (or be unspecified, which defaults to None). If this field is not specified, it is treated as an equivalent of Disabled.
`subPath`	`string`	Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).

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

.spec.resources.kubernetesResource.spec.template.spec.nodeSelector

Description

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

Type

`object`

.spec.resources.kubernetesResource.spec.template.spec.os

Description

Specifies the OS of the containers in the pod. Some pod and container fields are restricted if this is set. If the OS field is set to linux, the following fields must be unset: - securityContext.windowsOptions If the OS field is set to windows, following fields must be unset: - spec.hostPID - spec.hostIPC - spec.hostUsers - spec.resources - spec.securityContext.appArmorProfile - spec.securityContext.seLinuxOptions - spec.securityContext.seccompProfile - spec.securityContext.fsGroup - spec.securityContext.fsGroupChangePolicy - spec.securityContext.sysctls - spec.shareProcessNamespace - spec.securityContext.runAsUser -

spec.securityContext.runAsGroup - spec.securityContext.supplementalGroups - spec.securityContext.supplementalGroupsPolicy - spec.containers[*].securityContext.appArmorProfile - spec.containers[*].securityContext.seLinuxOptions - spec.containers[*].securityContext.seccompProfile - spec.containers[*].securityContext.capabilities - spec.containers[*].securityContext.readOnlyRootFilesystem - spec.containers[*].securityContext.privileged - spec.containers[*].securityContext.allowPrivilegeEscalation - spec.containers[*].securityContext.procMount - spec.containers[*].securityContext.runAsUser - spec.containers[*].securityContext.runAsGroup

Type

`object`

Required

`name`

Property	Type	Description
`name`	`string`	Name is the name of the operating system. The currently supported values are linux and windows. Additional value may be defined in future and can be one of: https://github.com/opencontainers/runtime-spec/blob/master/config.md#platform-specific-configuration ↗ Clients should expect to handle additional values and treat unrecognized values in this field as os: null

.spec.resources.kubernetesResource.spec.template.spec.overhead

Description

Overhead represents the resource overhead associated with running a pod for a given RuntimeClass. This field will be autopopulated at admission time by the RuntimeClass

admission controller. If the RuntimeClass admission controller is enabled, overhead must not be set in Pod create requests. The RuntimeClass admission controller will reject Pod create requests which have the overhead already set. If RuntimeClass is configured and selected in the PodSpec, Overhead will be set to the value defined in the corresponding RuntimeClass, otherwise it will remain unset and treated as zero. More info: https://git.k8s.io/enhancements/keps/sig-node/688-pod-overhead/README.md

Type

`object`

.spec.resources.kubernetesResource.spec.template.spec.readinessGates

Description

If specified, all readiness gates will be evaluated for pod readiness. A pod is ready when all its containers are ready AND all conditions specified in the readiness gates have status equal to "True" More info: https://git.k8s.io/enhancements/keps/sig-network/580-pod-readiness-gates

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.readinessGates[]

Description

PodReadinessGate contains the reference to a pod condition

Type

`object`

Required

`conditionType`

Property	Type	Description
`conditionType`	`string`	ConditionType refers to a condition in the pod's condition list with matching type.

.spec.resources.kubernetesResource.spec.template.spec.resourceClaims

Description

ResourceClaims defines which ResourceClaims must be allocated and reserved before the Pod is allowed to start. The resources will be made available to those containers which consume them by name. This is a stable field but requires that the DynamicResourceAllocation feature gate is enabled. This field is immutable.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.resourceClaims[]

Description

PodResourceClaim references exactly one ResourceClaim, either directly or by naming a ResourceClaimTemplate which is then turned into a ResourceClaim for the pod. It adds a name to it that uniquely identifies the ResourceClaim inside the Pod. Containers that need access to the ResourceClaim reference it with this name.

Type

`object`

Required

`name`

Property	Type	Description
`name`	`string`	Name uniquely identifies this resource claim inside the pod. This must be a DNS_LABEL.
`resourceClaimName`	`string`	ResourceClaimName is the name of a ResourceClaim object in the same namespace as this pod. Exactly one of ResourceClaimName and ResourceClaimTemplateName must be set.
`resourceClaimTemplateName`	`string`	ResourceClaimTemplateName is the name of a ResourceClaimTemplate object in the same namespace as this pod. The template will be used to create a new ResourceClaim, which will be bound to this pod. When this pod is deleted, the ResourceClaim will also be deleted. The pod name and resource name, along with a generated component, will be used to form a unique name for the ResourceClaim, which will be recorded in pod.status.resourceClaimStatuses. This field is immutable and no changes will be made to the corresponding ResourceClaim by the control plane after creating the ResourceClaim.

Property	Type	Description
		Exactly one of ResourceClaimName and ResourceClaimTemplateName must be set.

.spec.resources.kubernetesResource.spec.template.spec.resources

Description

Resources is the total amount of CPU and Memory resources required by all containers in the pod. It supports specifying Requests and Limits for "cpu", "memory" and "hugepages-" resource names only. ResourceClaims are not supported. This field enables fine-grained control over resource allocation for the entire pod, allowing resource sharing among containers in a pod. This is an alpha field and requires enabling the PodLevelResources feature gate.

Type

`object`

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

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.resources.kubernetesResource.spec.template.spec.resources.claims

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.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.resources.kubernetesResource.spec.template.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.kubernetesResource.spec.template.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.resources.kubernetesResource.spec.template.spec.schedulingGates

Description

SchedulingGates is an opaque list of values that if specified will block scheduling the pod. If schedulingGates is not empty, the pod will stay in the SchedulingGated state and the scheduler will not attempt to schedule the pod. SchedulingGates can only be set at pod creation time, and be removed only afterwards.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.schedulingGates[]

Description

PodSchedulingGate is associated to a Pod to guard its scheduling.

Type

`object`

Required

`name`

Property	Type	Description
`name`	`string`	Name of the scheduling gate. Each scheduling gate must have a unique name field.

.spec.resources.kubernetesResource.spec.template.spec.securityContext

Description

SecurityContext holds pod-level security attributes and common container settings. Optional: Defaults to empty. See type description for default values of each field.

Type

`object`

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

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

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

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

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

Property	Type	Description
		be enabled and the container runtime must implement support for this feature. Note that this field cannot be set when spec.os.name is windows.
`sysctls`	`array`	Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.
`windowsOptions`	`object`	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.resources.kubernetesResource.spec.template.spec.securityContext.appArmorProfile

Description

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

Type

`object`

Required

`type`

Property	Type	Description
`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.resources.kubernetesResource.spec.template.spec.securityContext.seLinuxOptions

Description

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

Type

`object`

Property	Type	Description
`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.resources.kubernetesResource.spec.template.spec.securityContext.seccompProfile

Description

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

Type

`object`

Required

`type`

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

.spec.resources.kubernetesResource.spec.template.spec.securityContext.supplementalGroups

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.securityContext.supplementalGroups[]

Type

`integer`

.spec.resources.kubernetesResource.spec.template.spec.securityContext.sysctls

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.securityContext.sysctls[]

Description

Sysctl defines a kernel parameter to be set

Type

`object`

Required

`name` `value`

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

.spec.resources.kubernetesResource.spec.template.spec.securityContext.windowsOptions

Description

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

Type

`object`

Property	Type	Description
`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

Property	Type	Description
		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.resources.kubernetesResource.spec.template.spec.tolerations

Description

If specified, the pod's tolerations.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.tolerations[]

Description

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

Type

`object`

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

Property	Type	Description
`value`	`string`	Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

.spec.resources.kubernetesResource.spec.template.spec.topologySpreadConstraints

Description

TopologySpreadConstraints describes how a group of pods ought to spread across topology domains. Scheduler will schedule pods in a way which abides by the constraints. All topologySpreadConstraints are ANDed.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.topologySpreadConstraints[]

Description

TopologySpreadConstraint specifies how to spread matching pods among the given topology.

Type

`object`

Required

`maxSkew` `topologyKey` `whenUnsatisfiable`

Property	Type	Description
`labelSelector`	`object`	LabelSelector is used to find matching pods. Pods that match this label selector are counted to determine the number of pods in their corresponding topology domain.
`matchLabelKeys`	`array`	MatchLabelKeys is a set of pod label keys to select the pods over which spreading will be calculated. The keys are used to lookup values from the incoming pod labels, those key-value labels are ANDed with labelSelector to select the group of existing pods over which spreading will be calculated for the incoming pod. The same key is forbidden to exist in both MatchLabelKeys and LabelSelector. MatchLabelKeys cannot be set when LabelSelector isn't set. Keys that don't exist in the incoming pod labels will be ignored. A null or empty list means only match against labelSelector. This is a beta field and requires the MatchLabelKeysInPodTopologySpread feature gate to be enabled (enabled by default).
`maxSkew`	`integer`	MaxSkew describes the degree to which pods may be unevenly distributed. When `whenUnsatisfiable=DoNotSchedule`, it is the maximum permitted difference between the number of matching pods in the target topology and the global minimum. The global minimum is the minimum number of matching pods in an eligible domain or zero if the number of eligible domains is less than MinDomains. For example, in

Property	Type	Description
		a 3-zone cluster, MaxSkew is set to 1, and pods with the same labelSelector spread as 2/2/1: In this case, the global minimum is 1. | zone1 | zone2 | zone3 | | P P | P P | P | • if MaxSkew is 1, incoming pod can only be scheduled to zone3 to become 2/2/2; scheduling it onto zone1(zone2) would make the ActualSkew(3-1) on zone1(zone2) violate MaxSkew(1). • if MaxSkew is 2, incoming pod can be scheduled onto any zone. When `whenUnsatisfiable=ScheduleAnyway`, it is used to give higher precedence to topologies that satisfy it. It's a required field. Default value is 1 and 0 is not allowed.
`minDomains`	`integer`	MinDomains indicates a minimum number of eligible domains. When the number of eligible domains with matching topology keys is less than minDomains, Pod Topology Spread treats "global minimum" as 0, and then the calculation of Skew is performed. And when the number of eligible domains with matching topology keys equals or greater than minDomains, this value has no effect on scheduling. As a result, when the number of eligible domains is less than minDomains, scheduler won't schedule more than maxSkew Pods to those domains. If value is nil, the constraint behaves as if MinDomains is equal to 1. Valid values are integers greater than 0. When value is not nil, WhenUnsatisfiable must be DoNotSchedule.

Property	Type	Description
		For example, in a 3-zone cluster, MaxSkew is set to 2, MinDomains is set to 5 and pods with the same labelSelector spread as 2/2/2: | zone1 | zone2 | zone3 | | P P | P P | P P | The number of domains is less than 5(MinDomains), so "global minimum" is treated as 0. In this situation, new pod with the same labelSelector cannot be scheduled, because computed skew will be 3(3 - 0) if new Pod is scheduled to any of the three zones, it will violate MaxSkew.
`nodeAffinityPolicy`	`string`	NodeAffinityPolicy indicates how we will treat Pod's nodeAffinity/nodeSelector when calculating pod topology spread skew. Options are: • Honor: only nodes matching nodeAffinity/nodeSelector are included in the calculations. • Ignore: nodeAffinity/nodeSelector are ignored. All nodes are included in the calculations. If this value is nil, the behavior is equivalent to the Honor policy.
`nodeTaintsPolicy`	`string`	NodeTaintsPolicy indicates how we will treat node taints when calculating pod topology spread skew. Options are: • Honor: nodes without taints, along with tainted nodes for which the incoming pod has a toleration, are included. • Ignore: node taints are ignored. All nodes are included.

Property	Type	Description
		If this value is nil, the behavior is equivalent to the Ignore policy.
`topologyKey`	`string`	TopologyKey is the key of node labels. Nodes that have a label with this key and identical values are considered to be in the same topology. We consider each <key, value> as a "bucket", and try to put balanced number of pods into each bucket. We define a domain as a particular instance of a topology. Also, we define an eligible domain as a domain whose nodes meet the requirements of nodeAffinityPolicy and nodeTaintsPolicy. e.g. If TopologyKey is "kubernetes.io/hostname", each Node is a domain of that topology. And, if TopologyKey is "topology.kubernetes.io/zone", each zone is a domain of that topology. It's a required field.
`whenUnsatisfiable`	`string`	WhenUnsatisfiable indicates how to deal with a pod if it doesn't satisfy the spread constraint. • DoNotSchedule (default) tells the scheduler not to schedule it. • ScheduleAnyway tells the scheduler to schedule the pod in any location, but giving higher precedence to topologies that would help reduce the skew. A constraint is considered "Unsatisfiable" for an incoming pod if and only if every possible node assignment for that pod would violate "MaxSkew" on some topology. For example, in a 3-zone cluster, MaxSkew is set to 1, and pods with the same

Property	Type	Description
		labelSelector spread as 3/1/1: | zone1 | zone2 | zone3 | | P P P | P | P | If WhenUnsatisfiable is set to DoNotSchedule, incoming pod can only be scheduled to zone2(zone3) to become 3/2/1(3/1/2) as ActualSkew(2-1) on zone2(zone3) satisfies MaxSkew(1). In other words, the cluster can still be imbalanced, but scheduler won't make it *more* imbalanced. It's a required field.

.spec.resources.kubernetesResource.spec.template.spec.topologySpreadConstraints[].labelSelector

Description

LabelSelector is used to find matching pods. Pods that match this label selector are counted to determine the number of pods in their corresponding topology domain.

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.resources.kubernetesResource.spec.template.spec.topologySpreadConstraints[].labelSelector.matchExpressions

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.topologySpreadConstraints[].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.resources.kubernetesResource.spec.template.spec.topologySpreadConstraints[].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.resources.kubernetesResource.spec.template.spec.topologySpreadConstraints[].labelSelector.matchExpressions[].values[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.topologySpreadConstraints[].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.resources.kubernetesResource.spec.template.spec.topologySpreadConstraints[].matchLabelKeys

Description

MatchLabelKeys is a set of pod label keys to select the pods over which spreading will be calculated. The keys are used to lookup values from the incoming pod labels, those key-value labels are ANDed with labelSelector to select the group of existing pods over which spreading will be calculated for the incoming pod. The same key is forbidden to exist in both MatchLabelKeys and LabelSelector. MatchLabelKeys cannot be set when LabelSelector isn't set. Keys that don't exist in the incoming pod labels will be ignored. A null or empty list means only match against labelSelector. This is a beta field and requires the MatchLabelKeysInPodTopologySpread feature gate to be enabled (enabled by default).

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.topologySpreadConstraints[].matchLabelKeys[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.volumes

Description

List of volumes that can be mounted by containers belonging to the pod. More info: https://kubernetes.io/docs/concepts/storage/volumes

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.volumes[]

Description

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

Type

`object`

Required

`name`

Property	Type	Description
`awsElasticBlockStore`	`object`	awsElasticBlockStore represents an AWS Disk reso a kubelet's host machine and then exposed to the p AWSElasticBlockStore is deprecated. All operations awsElasticBlockStore type are redirected to the ebs. driver. More info: https://kubernetes.io/docs/concepts/storage/volume ↗
`azureDisk`	`object`	azureDisk represents an Azure Data Disk mount on mount to the pod. Deprecated: AzureDisk is deprec the in-tree azureDisk type are redirected to the disk. driver.
`azureFile`	`object`	azureFile represents an Azure File Service mount o mount to the pod. Deprecated: AzureFile is deprec

Property	Type	Description
		the in-tree azureFile type are redirected to the file.cs driver.
`cephfs`	`object`	cephFS represents a Ceph FS mount on the host th lifetime. Deprecated: CephFS is deprecated and the no longer supported.
`cinder`	`object`	cinder represents a cinder volume attached and mou machine. Deprecated: Cinder is deprecated. All oper cinder type are redirected to the cinder.csi.openstac info: https://examples.k8s.io/mysql-cinder-pd/READI
`configMap`	`object`	configMap represents a configMap that should popu
`csi`	`object`	csi (Container Storage Interface) represents epheme handled by certain external CSI drivers.
`downwardAPI`	`object`	downwardAPI represents downward API about the p populate this volume
`emptyDir`	`object`	emptyDir represents a temporary directory that shar More info: https://kubernetes.io/docs/concepts/stora ↗
`ephemeral`	`object`	ephemeral represents a volume that is handled by a The volume's lifecycle is tied to the pod that defines

Property	Type	Description
		before the pod starts, and deleted when the pod is r Use this if: a) the volume is only needed while the p normal volumes like restoring from snapshot or capa needed, c) the storage driver is specified through a s the storage driver supports dynamic volume provisio PersistentVolumeClaim (see EphemeralVolumeSour information on the connection between this volume t PersistentVolumeClaim). Use PersistentVolumeClaim or one of the vendor-sp volumes that persist for longer than the lifecycle of a Use CSI for light-weight local ephemeral volumes if t meant to be used that way - see the documentation information. A pod can use both types of ephemeral volumes and at the same time.
`fc`	`object`	fc represents a Fibre Channel resource that is attach machine and then exposed to the pod.
`flexVolume`	`object`	flexVolume represents a generic volume resource th provisioned/attached using an exec based plugin. D FlexVolume is deprecated. Consider using a CSIDriv
`flocker`	`object`	flocker represents a Flocker volume attached to a ku This depends on the Flocker control service being ru Flocker is deprecated and the in-tree flocker type is

Property	Type	Description
`gcePersistentDisk`	`object`	gcePersistentDisk represents a GCE Disk resource kubelet's host machine and then exposed to the pod GCEPersistentDisk is deprecated. All operations for gcePersistentDisk type are redirected to the pd.csi.s driver. More info: https://kubernetes.io/docs/concepts/storage/volumes
`gitRepo`	`object`	gitRepo represents a git repository at a particular rev GitRepo is deprecated. To provision a container with EmptyDir into an InitContainer that clones the repo u the EmptyDir into the Pod's container.
`glusterfs`	`object`	glusterfs represents a Glusterfs mount on the host th lifetime. Deprecated: Glusterfs is deprecated and the is no longer supported.
`hostPath`	`object`	hostPath represents a pre-existing file or directory o that is directly exposed to the container. This is gene agents or other privileged things that are allowed to Most containers will NOT need this. More info: https://kubernetes.io/docs/concepts/storage/volumes
`image`	`object`	image represents an OCI object (a container image mounted on the kubelet's host machine. The volume startup depending on which PullPolicy value is provi • Always: the kubelet always attempts to pull the re creation will fail If the pull fails.

Property	Type	Description
		• Never: the kubelet never pulls the reference and image or artifact. Container creation will fail if the present. • IfNotPresent: the kubelet pulls if the reference isr disk. Container creation will fail if the reference is pull fails. The volume gets re-resolved if the pod gets deleted means that new remote content will become availab failure to resolve or pull the image during pod startu from starting and may add significant latency. Failure normal volume backoff and will be reported on the p message. The types of objects that may be mounted defined by the container runtime implementation on minimum must include all valid types supported by tl field. The OCI object gets mounted in a single direct (spec.containers[*].volumeMounts.mountPath) by me layers in the same way as for container images. The mounted read-only (ro) and non-executable files (no mounts for containers are not supported (spec.containers[].volumeMounts.subpath) before 1. spec.securityContext.fsGroupChangePolicy has no type.
`iscsi`	`object`	iscsi represents an ISCSI Disk resource that is attac machine and then exposed to the pod. More info: https://kubernetes.io/docs/concepts/storage/volumes
`name`	`string`	name of the volume. Must be a DNS_LABEL and un More info: https://kubernetes.io/docs/concepts/overv objects/names/#names ↗

Property	Type	Description
`nfs`	`object`	nfs represents an NFS mount on the host that share info: https://kubernetes.io/docs/concepts/storage/vol
`persistentVolumeClaim`	`object`	persistentVolumeClaimVolumeSource represents a PersistentVolumeClaim in the same namespace. Mo https://kubernetes.io/docs/concepts/storage/persiste volumes#persistentvolumeclaims ↗
`photonPersistentDisk`	`object`	photonPersistentDisk represents a PhotonController attached and mounted on kubelets host machine. De PhotonPersistentDisk is deprecated and the in-tree type is no longer supported.
`portworxVolume`	`object`	portworxVolume represents a portworx volume attac kubelets host machine. Deprecated: PortworxVolum operations for the in-tree portworxVolume type are r pxd.portworx.com CSI driver when the CSIMigration is on.
`projected`	`object`	projected items for all in one resources secrets, conf downward API
`quobyte`	`object`	quobyte represents a Quobyte mount on the host th lifetime. Deprecated: Quobyte is deprecated and the is no longer supported.

Property	Type	Description
`rbd`	`object`	rbd represents a Rados Block Device mount on the pod's lifetime. Deprecated: RBD is deprecated and t no longer supported.
`scaleIO`	`object`	scaleIO represents a ScaleIO persistent volume atta Kubernetes nodes. Deprecated: ScaleIO is deprecat scaleIO type is no longer supported.
`secret`	`object`	secret represents a secret that should populate this https://kubernetes.io/docs/concepts/storage/volumes
`storageos`	`object`	storageOS represents a StorageOS volume attache Kubernetes nodes. Deprecated: StorageOS is depre storageos type is no longer supported.
`vsphereVolume`	`object`	vsphereVolume represents a vSphere volume attach kubelets host machine. Deprecated: VsphereVolume operations for the in-tree vsphereVolume type are re csi.vsphere.vmware.com CSI driver.

.spec.resources.kubernetesResource.spec.template.spec.volumes[].awsElasticBlockStore

Description

awsElasticBlockStore represents an AWS Disk resource that is attached to a kubelet's host machine and then exposed to the pod. Deprecated: AWSElasticBlockStore is deprecated.

All operations for the in-tree awsElasticBlockStore type are redirected to the ebs.csi.aws.com CSI driver. More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

Type

`object`

Required

`volumeID`

Property	Type	Description
`fsType`	`string`	fsType is the filesystem type of the volume that you want to mou Ensure that the filesystem type is supported by the host operatir Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if ur More info: https://kubernetes.io/docs/concepts/storage/volumes#awselastic ↗
`partition`	`integer`	partition is the partition in the volume that you want to mount. If the default is to mount by volume name. Examples: For volume you specify the partition as "1". Similarly, the volume partition for is "0" (or you can leave the property empty).
`readOnly`	`boolean`	readOnly value true will force the readOnly setting in VolumeMo info: https://kubernetes.io/docs/concepts/storage/volumes#awselastic ↗

Property	Type	Description
`volumeID`	`string`	volumeID is unique ID of the persistent disk resource in AWS (A EBS volume). More info: https://kubernetes.io/docs/concepts/storage/volumes#awselastic ↗

.spec.resources.kubernetesResource.spec.template.spec.volumes[].azureDisk

Description

azureDisk represents an Azure Data Disk mount on the host and bind mount to the pod. Deprecated: AzureDisk is deprecated. All operations for the in-tree azureDisk type are redirected to the disk.csi.azure.com CSI driver.

Type

`object`

Required

`diskName` `diskURI`

Property	Type	Description
`cachingMode`	`string`	cachingMode is the Host Caching mode: None, Read Only, Read Write.
`diskName`	`string`	diskName is the Name of the data disk in the blob storage
`diskURI`	`string`	diskURI is the URI of data disk in the blob storage

Property	Type	Description
`fsType`	`string`	fsType is Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
`kind`	`string`	kind expected values are Shared: multiple blob disks per storage account Dedicated: single blob disk per storage account Managed: azure managed data disk (only in managed availability set). defaults to shared
`readOnly`	`boolean`	readOnly Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

.spec.resources.kubernetesResource.spec.template.spec.volumes[].azureFile

Description

azureFile represents an Azure File Service mount on the host and bind mount to the pod. Deprecated: AzureFile is deprecated. All operations for the in-tree azureFile type are redirected to the file.csi.azure.com CSI driver.

Type

`object`

Required

`secretName` `shareName`

Property	Type	Description
`readOnly`	`boolean`	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
`secretName`	`string`	secretName is the name of secret that contains Azure Storage Account Name and Key
`shareName`	`string`	shareName is the azure share Name

.spec.resources.kubernetesResource.spec.template.spec.volumes[].cephfs

Description

cephFS represents a Ceph FS mount on the host that shares a pod's lifetime. Deprecated: CephFS is deprecated and the in-tree cephfs type is no longer supported.

Type

`object`

Required

`monitors`

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

Property	Type	Description
`path`	`string`	path is Optional: Used as the mounted root, rather than the full Ceph tree, default is /
`readOnly`	`boolean`	readOnly is Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts. More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗
`secretFile`	`string`	secretFile is Optional: SecretFile is the path to key ring for User, default is /etc/ceph/user.secret More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗
`secretRef`	`object`	secretRef is Optional: SecretRef is reference to the authentication secret for User, default is empty. More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗
`user`	`string`	user is optional: User is the rados user name, default is admin More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it ↗

.spec.resources.kubernetesResource.spec.template.spec.volumes[].cephfs.monitors

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.volumes[].cephfs.monitors[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.volumes[].cephfs.secretRef

Description

secretRef is Optional: SecretRef is reference to the authentication secret for User, default is empty. More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it

Type

`object`

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

.spec.resources.kubernetesResource.spec.template.spec.volumes[].cinder

Description

cinder represents a cinder volume attached and mounted on kubelets host machine. Deprecated: Cinder is deprecated. All operations for the in-tree cinder type are redirected to the cinder.csi.openstack.org CSI driver. More info: https://examples.k8s.io/mysql-cinder-pd/README.md

Type

`object`

Required

`volumeID`

Property	Type	Description
`fsType`	`string`	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://examples.k8s.io/mysql-cinder-pd/README.md ↗
`readOnly`	`boolean`	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts. More info: https://examples.k8s.io/mysql-cinder-pd/README.md ↗
`secretRef`	`object`	secretRef is optional: points to a secret object containing parameters used to connect to OpenStack.

Property	Type	Description
`volumeID`	`string`	volumeID used to identify the volume in cinder. More info: https://examples.k8s.io/mysql-cinder-pd/README.md ↗

.spec.resources.kubernetesResource.spec.template.spec.volumes[].cinder.secretRef

Description

secretRef is optional: points to a secret object containing parameters used to connect to OpenStack.

Type

`object`

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

.spec.resources.kubernetesResource.spec.template.spec.volumes[].configMap

Description

configMap represents a configMap that should populate this volume

Type

`object`

Property	Type	Description
`defaultMode`	`integer`	defaultMode is optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
`items`	`array`	items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
`name`	`string`	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names↗
`optional`	`boolean`	optional specify whether the ConfigMap or its keys must be defined

.spec.resources.kubernetesResource.spec.template.spec.volumes[].configMap.items

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.volumes[].configMap.items[]

Description

Maps a string key to a path within a volume.

Type

`object`

Required

`key` `path`

Property	Type	Description
`key`	`string`	key is the key to project.
`mode`	`integer`	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values

Property	Type	Description
		for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
`path`	`string`	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

.spec.resources.kubernetesResource.spec.template.spec.volumes[].csi

Description

csi (Container Storage Interface) represents ephemeral storage that is handled by certain external CSI drivers.

Type

`object`

Required

`driver`

Property	Type	Description
`driver`	`string`	driver is the name of the CSI driver that handles this volume. Consult with your admin for the correct name as registered in the cluster.
`fsType`	`string`	fsType to mount. Ex. "ext4", "xfs", "ntfs". If not provided, the empty value is passed to the

Property	Type	Description
		associated CSI driver which will determine the default filesystem to apply.
`nodePublishSecretRef`	`object`	nodePublishSecretRef is a reference to the secret object containing sensitive information to pass to the CSI driver to complete the CSI NodePublishVolume and NodeUnpublishVolume calls. This field is optional, and may be empty if no secret is required. If the secret object contains more than one secret, all secret references are passed.
`readOnly`	`boolean`	readOnly specifies a read-only configuration for the volume. Defaults to false (read/write).
`volumeAttributes`	`object`	volumeAttributes stores driver-specific properties that are passed to the CSI driver. Consult your driver's documentation for supported values.

.spec.resources.kubernetesResource.spec.template.spec.volumes[].csi.nodePublishSecretRef

Description

nodePublishSecretRef is a reference to the secret object containing sensitive information to pass to the CSI driver to complete the CSI NodePublishVolume and NodeUnpublishVolume calls. This field is optional, and may be empty if no secret is required. If the secret object contains more than one secret, all secret references are passed.

Type

`object`

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

.spec.resources.kubernetesResource.spec.template.spec.volumes[].csi.volumeAttributes

Description

volumeAttributes stores driver-specific properties that are passed to the CSI driver. Consult your driver's documentation for supported values.

Type

`object`

.spec.resources.kubernetesResource.spec.template.spec.volumes[].downwardAPI

Description

downwardAPI represents downward API about the pod that should populate this volume

Type

`object`

Property	Type	Description
`defaultMode`	`integer`	Optional: mode bits to use on created files by default. Must be a Optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
`items`	`array`	Items is a list of downward API volume file

.spec.resources.kubernetesResource.spec.template.spec.volumes[].downwardAPI.items

Description

Items is a list of downward API volume file

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.volumes[].downwardAPI.items[]

Description

DownwardAPIVolumeFile represents information to create the file containing the pod field

Type

`object`

Required

`path`

Property	Type	Description
`fieldRef`	`object`	Required: Selects a field of the pod: only annotations, labels, name, namespace and uid are supported.
`mode`	`integer`	Optional: mode bits used to set permissions on this file, must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
`path`	`string`	Required: Path is the relative path name of the file to be created. Must not be absolute or contain the '..' path. Must be utf-8 encoded. The first item of the relative path must not start with '..'
`resourceFieldRef`	`object`	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, requests.cpu and requests.memory) are currently supported.

.spec.resources.kubernetesResource.spec.template.spec.volumes[].downwardAPI.items[].fieldRef

Description

Required: Selects a field of the pod: only annotations, labels, name, namespace and uid are supported.

Type

`object`

Required

`fieldPath`

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

.spec.resources.kubernetesResource.spec.template.spec.volumes[].downwardAPI.items[].resourceFieldRef

Description

Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, requests.cpu and requests.memory) are currently supported.

Type

`object`

Required

`resource`

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

.spec.resources.kubernetesResource.spec.template.spec.volumes[].emptyDir

Description

emptyDir represents a temporary directory that shares a pod's lifetime. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir

Type

object

Property	Type	Description
medium	string	medium represents what type of storage medium should back this directory. The default is "" which means to use the node's default medium. Must be an empty string (default) or Memory. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir ↗

Property	Type	Description
`sizeLimit`		sizeLimit is the total amount of local storage required for this EmptyDir volume. The size limit is also applicable for memory medium. The maximum usage on memory medium EmptyDir would be the minimum value between the SizeLimit specified here and the sum of memory limits of all containers in a pod. The default is nil which means that the limit is undefined. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir ↗

.spec.resources.kubernetesResource.spec.template.spec.volumes[].ephemeral

Description

ephemeral represents a volume that is handled by a cluster storage driver. The volume's lifecycle is tied to the pod that defines it - it will be created before the pod starts, and deleted when the pod is removed. Use this if: a) the volume is only needed while the pod runs, b) features of normal volumes like restoring from snapshot or capacity tracking are needed, c) the storage driver is specified through a storage class, and d) the storage driver supports dynamic volume provisioning through a PersistentVolumeClaim (see EphemeralVolumeSource for more information on the connection between this volume type and PersistentVolumeClaim). Use PersistentVolumeClaim or one of the vendor-specific APIs for volumes that persist for longer than the lifecycle of an individual pod. Use CSI for light-weight local ephemeral volumes if the CSI driver is meant to be used that way - see the documentation of the driver for more information. A pod can use both types of ephemeral volumes and persistent volumes at the same time.

Type

`object`

Property	Type	Description
`volumeClaimTemplate`	`object`	Will be used to create a stand-alone PVC to provision the volume. The pod in which this EphemeralVolumeSource is embedded will be the owner of the PVC, i.e. the PVC will be deleted together with the pod. The name of the PVC will be `<pod name>-<volume name>` where `<volume name>` is the name from the `PodSpec.Volumes` array entry. Pod validation will reject the pod if the concatenated name is not valid for a PVC (for example, too long). An existing PVC with that name that is not owned by the pod will *not* be used for the pod to avoid using an unrelated volume by mistake. Starting the pod is then blocked until the unrelated PVC is removed. If such a pre-created PVC is meant to be used by the pod, the PVC has to updated with an owner reference to the pod once the pod exists. Normally this should not be necessary, but it may be useful when manually reconstructing a broken cluster. This field is read-only and no changes will be made by Kubernetes to the PVC after it has been created. Required, must not be nil.

.spec.resources.kubernetesResource.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate

Description

Will be used to create a stand-alone PVC to provision the volume. The pod in which this EphemeralVolumeSource is embedded will be the owner of the PVC, i.e. the PVC will be deleted together with the pod. The name of the PVC will be `<pod name>-<volume name>` where `<volume name>` is the name from the `PodSpec.Volumes` array entry. Pod validation will reject the pod if the concatenated name is not valid for a PVC (for example, too long). An existing PVC with that name that is not owned by the pod will *not* be used for the pod to avoid using an unrelated volume by mistake. Starting the pod is then blocked until the unrelated PVC is removed. If such a pre-created PVC is meant to be used by the pod, the PVC has to updated with an owner reference to the pod once the pod exists. Normally this should not be necessary, but it may be useful when manually reconstructing a broken cluster. This field is read-only and no changes will be made by Kubernetes to the PVC after it has been created. Required, must not be nil.

Type

`object`

Required

`spec`

Property	Type	Description
`metadata`	`ObjectMeta`	May contain labels and annotations that will be copied into the PVC when creating it. No other fields are allowed and will be rejected during validation.
`spec`	`object`	The specification for the PersistentVolumeClaim. The entire content is copied unchanged into the PVC that gets created from this template. The same fields as in a PersistentVolumeClaim are also valid here.

.spec.resources.kubernetesResource.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec

Description

The specification for the PersistentVolumeClaim. The entire content is copied unchanged into the PVC that gets created from this template. The same fields as in a PersistentVolumeClaim are also valid here.

Type

`object`

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

Property	Type	Description
		PersistentVolumeClaim object. When this field i specified, volume binding will only succeed if th of the specified object matches some installed v populator or dynamic provisioner. This field will the functionality of the dataSource field and as s both fields are non-empty, they must have the s value. For backwards compatibility, when name isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to t same value automatically if one of them is empt the other is non-empty. When namespace is sp in dataSourceRef, dataSource isn't set to the sa value and must be empty. There are three impo differences between dataSource and dataSourc • While dataSource only allows two specific ty objects, dataSourceRef allows any non-core as well as PersistentVolumeClaim objects. • While dataSource ignores disallowed values (dropping them), dataSourceRef preserves a values, and generates an error if a disallowe value is specified. • While dataSource only allows local objects, dataSourceRef allows objects in any names (Beta) Using this field requires the AnyVolumeDataSource feature gate to be en (Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource featur to be enabled.
`resources`	`object`	resources represents the minimum resources th volume should have. Users are allowed to spec resource requirements that are lower than previ

Property	Type	Description
		value but must still be higher than capacity recc the status field of the claim. More info: https://kubernetes.io/docs/concepts/storage/per volumes#resources ↗
`selector`	`object`	selector is a label query over volumes to consid binding.
`storageClassName`	`string`	storageClassName is the name of the StorageC required by the claim. More info: https://kubernetes.io/docs/concepts/storage/per volumes#class-1 ↗
`volumeAttributesClassName`	`string`	volumeAttributesClassName may be used to se VolumeAttributesClass used by this claim. If spe the CSI driver will create or update the volume attributes defined in the corresponding VolumeAttributesClass. This has a different pur than storageClassName, it can be changed afte claim is created. An empty string or nil value inc that no VolumeAttributesClass will be applied tc claim. If the claim enters an Infeasible error stat field can be reset to its previous value (includin cancel the modification. If the resource referred volumeAttributesClass does not exist, this PersistentVolumeClaim will be set to a Pending as reflected by the modifyVolumeStatus field, u such as a resource exists. More info: https://kubernetes.io/docs/concepts/storage/vol attributes-classes/ ↗

Property	Type	Description
`volumeMode`	`string`	volumeMode defines what type of volume is rec by the claim. Value of Filesystem is implied whe included in claim spec.
`volumeName`	`string`	volumeName is the binding reference to the PersistentVolume backing this claim.

.spec.resources.kubernetesResource.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.accessModes

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.accessModes[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSou

rce

Description

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

Type

`object`

Required

`kind` `name`

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

.spec.resources.kubernetesResource.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSou

rceRef

Description

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

Type

`object`

Required

`kind` `name`

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

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

.spec.resources.kubernetesResource.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources

Description

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

Type

`object`

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.resources.kubernetesResource.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources.limits

Description

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

Type

`object`

.spec.resources.kubernetesResource.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources.requests

Description

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

Type

`object`

.spec.resources.kubernetesResource.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector

Description

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

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.resources.kubernetesResource.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions[]

Description

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

Type

`object`

Required

`key` `operator`

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

.spec.resources.kubernetesResource.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions[].values

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchExpressions[].values[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.selector.matchLabels

Description

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

Type

`object`

.spec.resources.kubernetesResource.spec.template.spec.volumes[].fc

Description

fc represents a Fibre Channel resource that is attached to a kubelet's host machine and then exposed to the pod.

Type

`object`

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

.spec.resources.kubernetesResource.spec.template.spec.volumes[].fc.targetWWNs

Description

targetWWNs is Optional: FC target worldwide names (WWNs)

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.volumes[].fc.targetWWNs[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.volumes[].fc.wwids

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.volumes[].fc.wwids[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.volumes[].flexVolume

Description

flexVolume represents a generic volume resource that is provisioned/attached using an exec based plugin. Deprecated: FlexVolume is deprecated. Consider using a CSIDriver instead.

Type

`object`

Required

`driver`

Property	Type	Description
`driver`	`string`	driver is the name of the driver to use for this volume.
`fsType`	`string`	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". The default filesystem depends on FlexVolume script.
`options`	`object`	options is Optional: this field holds extra command options if any.
`readOnly`	`boolean`	readOnly is Optional: defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

Property	Type	Description
`secretRef`	`object`	secretRef is Optional: secretRef is reference to the secret object containing sensitive information to pass to the plugin scripts. This may be empty if no secret object is specified. If the secret object contains more than one secret, all secrets are passed to the plugin scripts.

.spec.resources.kubernetesResource.spec.template.spec.volumes[].flexVolume.options

Description

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

Type

`object`

.spec.resources.kubernetesResource.spec.template.spec.volumes[].flexVolume.secretRef

Description

secretRef is Optional: secretRef is reference to the secret object containing sensitive information to pass to the plugin scripts. This may be empty if no secret object is specified. If the secret object contains more than one secret, all secrets are passed to the plugin scripts.

Type

`object`

Property	Type	Description
`name`	`string`	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty.

Property	Type	Description
		Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

.spec.resources.kubernetesResource.spec.template.spec.volumes[].flocker

Description

flocker represents a Flocker volume attached to a kubelet's host machine. This depends on the Flocker control service being running. Deprecated: Flocker is deprecated and the in-tree flocker type is no longer supported.

Type

`object`

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

.spec.resources.kubernetesResource.spec.template.spec.volumes[].gcePersistentDisk

Description

gcePersistentDisk represents a GCE Disk resource that is attached to a kubelet's host machine and then exposed to the pod. Deprecated: GCEPersistentDisk is deprecated. All operations for the in-tree gcePersistentDisk type are redirected to the pd.csi.storage.gke.io CSI driver. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk

Type

`object`

Required

`pdName`

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

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

.spec.resources.kubernetesResource.spec.template.spec.volumes[].gitRepo

Description

gitRepo represents a git repository at a particular revision. Deprecated: GitRepo is deprecated. To provision a container with a git repo, mount an EmptyDir into an InitContainer that clones the repo using git, then mount the EmptyDir into the Pod's container.

Type

`object`

Required

`repository`

Property	Type	Description
`directory`	`string`	directory is the target directory name. Must not contain or start with '..'. If '.' is supplied, the volume directory will be the git repository. Otherwise, if specified, the volume will contain the git repository in the subdirectory with the given name.
`repository`	`string`	repository is the URL

Property	Type	Description
`revision`	`string`	revision is the commit hash for the specified revision.

.spec.resources.kubernetesResource.spec.template.spec.volumes[].glusterfs

Description

glusterfs represents a Glusterfs mount on the host that shares a pod's lifetime. Deprecated: Glusterfs is deprecated and the in-tree glusterfs type is no longer supported.

Type

`object`

Required

`endpoints` `path`

Property	Type	Description
`endpoints`	`string`	endpoints is the endpoint name that details Glusterfs topology.
`path`	`string`	path is the Glusterfs volume path. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod ↗
`readOnly`	`boolean`	readOnly here will force the Glusterfs volume to be mounted with read-only permissions. Defaults to false. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod ↗

.spec.resources.kubernetesResource.spec.template.spec.volumes[].hostPath

Description

hostPath represents a pre-existing file or directory on the host machine that is directly exposed to the container. This is generally used for system agents or other privileged things that are allowed to see the host machine. Most containers will NOT need this. More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath

Type

`object`

Required

`path`

Property	Type	Description
`path`	`string`	path of the directory on the host. If the path is a symlink, it will follow the link to the real path. More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath ↗
`type`	`string`	type for HostPath Volume Defaults to "" More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath ↗

.spec.resources.kubernetesResource.spec.template.spec.volumes[].image

Description

image represents an OCI object (a container image or artifact) pulled and mounted on the kubelet's host machine. The volume is resolved at pod startup depending on which PullPolicy value is provided: - Always: the kubelet always attempts to pull the reference.

Container creation will fail If the pull fails. - Never: the kubelet never pulls the reference and only uses a local image or artifact. Container creation will fail if the reference isn't present. - IfNotPresent: the kubelet pulls if the reference isn't already present on disk. Container creation will fail if the reference isn't present and the pull fails. The volume gets re-resolved if the pod gets deleted and recreated, which means that new remote content will become available on pod recreation. A failure to resolve or pull the image during pod startup will block containers from starting and may add significant latency. Failures will be retried using normal volume backoff and will be reported on the pod reason and message. The types of objects that may be mounted by this volume are defined by the container runtime implementation on a host machine and at minimum must include all valid types supported by the container image field. The OCI object gets mounted in a single directory (spec.containers[*].volumeMounts.mountPath) by merging the manifest layers in the same way as for container images. The volume will be mounted read-only (ro) and non-executable files (noexec). Sub path mounts for containers are not supported (spec.containers[*].volumeMounts.subpath) before 1.33. The field spec.securityContext.fsGroupChangePolicy has no effect on this volume type.

Type

`object`

Property	Type	Description
`pullPolicy`	`string`	Policy for pulling OCI objects. Possible values are: Always: the kubelet always attempts to pull the reference. Container creation will fail If the pull fails. Never: the kubelet never pulls the reference and only uses a local image or artifact. Container creation will fail if the reference isn't present. IfNotPresent: the kubelet pulls if the reference isn't already present on disk. Container creation will fail if the reference isn't present and the pull fails. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise.
`reference`	`string`	Required: Image or artifact reference to be used. Behaves in the same way as pod.spec.containers[*].image. Pull secrets will be assembled in the same way as for the container image by looking up node credentials, SA image

Property	Type	Description
		pull secrets, and pod spec image pull secrets. More info: https://kubernetes.io/docs/concepts/containers/images This field is optional to allow higher level config management to default or override container images in workload controllers like Deployments and StatefulSets.

.spec.resources.kubernetesResource.spec.template.spec.volumes[].iscsi

Description

iscsi represents an ISCSI Disk resource that is attached to a kubelet's host machine and then exposed to the pod. More info: https://kubernetes.io/docs/concepts/storage/volumes/#iscsi

Type

`object`

Required

`iqn` `lun` `targetPortal`

Property	Type	Description
`chapAuthDiscovery`	`boolean`	chapAuthDiscovery defines whether support iSCSI Discovery CHAP authentication
`chapAuthSession`	`boolean`	chapAuthSession defines whether support iSCSI Sessior CHAP authentication
`fsType`	`string`	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supporte

Property	Type	Description
		by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#isc ↗
`initiatorName`	`string`	initiatorName is the custom iSCSI Initiator Name. If initiatorName is specified with iscsiInterface simultaneously, new iSCSI interface : will be created for the connection.
`iqn`	`string`	iqn is the target iSCSI Qualified Name.
`iscsiInterface`	`string`	iscsiInterface is the interface Name that uses an iSCSI transport. Defaults to 'default' (tcp).
`lun`	`integer`	lun represents iSCSI Target Lun number.
`portals`	`array`	portals is the iSCSI Target Portal List. The portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).
`readOnly`	`boolean`	readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false.

Property	Type	Description
`secretRef`	`object`	secretRef is the CHAP Secret for iSCSI target and initiatc authentication
`targetPortal`	`string`	targetPortal is iSCSI Target Portal. The Portal is either ar IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).

.spec.resources.kubernetesResource.spec.template.spec.volumes[].iscsi.portals

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.volumes[].iscsi.portals[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.volumes[].iscsi.secretRef

Description

secretRef is the CHAP Secret for iSCSI target and initiator authentication

Type

`object`

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

.spec.resources.kubernetesResource.spec.template.spec.volumes[].nfs

Description

nfs represents an NFS mount on the host that shares a pod's lifetime More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs

Type

`object`

Required

`path` `server`

Property	Type	Description
`path`	`string`	path that is exported by the NFS server. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs ↗
`readOnly`	`boolean`	readOnly here will force the NFS export to be mounted with read-only permissions. Defaults to false. More info:

Property	Type	Description
		https://kubernetes.io/docs/concepts/storage/volumes#nfs ↗
`server`	`string`	server is the hostname or IP address of the NFS server. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs ↗

.spec.resources.kubernetesResource.spec.template.spec.volumes[].persistentVolumeClaim

Description

persistentVolumeClaimVolumeSource represents a reference to a PersistentVolumeClaim in the same namespace. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims

Type

`object`

Required

`claimName`

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

.spec.resources.kubernetesResource.spec.template.spec.volumes[].photonPersistentDisk

Description

photonPersistentDisk represents a PhotonController persistent disk attached and mounted on kubelets host machine. Deprecated: PhotonPersistentDisk is deprecated and the in-tree photonPersistentDisk type is no longer supported.

Type

`object`

Required

`pdID`

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

.spec.resources.kubernetesResource.spec.template.spec.volumes[].portworxVolume

Description

portworxVolume represents a portworx volume attached and mounted on kubelets host machine. Deprecated: PortworxVolume is deprecated. All operations for the in-tree portworxVolume type are redirected to the pxd.portworx.com CSI driver when the CSIMigrationPortworx feature-gate is on.

Type

`object`

Required

`volumeID`

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

.spec.resources.kubernetesResource.spec.template.spec.volumes[].projected

Description

projected items for all in one resources secrets, configmaps, and downward API

Type

`object`

Property	Type	Description
`defaultMode`	`integer`	defaultMode are the mode bits used to set permissions on created files by default. Must be an octal value between

Property	Type	Description
		0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
`sources`	`array`	sources is the list of volume projections. Each entry in this list handles one source.

.spec.resources.kubernetesResource.spec.template.spec.volumes[].projected.sources

Description

sources is the list of volume projections. Each entry in this list handles one source.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.volumes[].projected.sources[]

Description

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

Type

`object`

Property	Type	Description
`clusterTrustBundle`	`object`	ClusterTrustBundle allows a pod to access the `.spec.trustBundle` field of ClusterTrustBundle objects in an auto-updating file. Alpha, gated by the ClusterTrustBundleProjection feature gate. ClusterTrustBundle objects can either be selected by name, or by the combination of signer name and a label selector. Kubelet performs aggressive normalization of the PEM contents written into the pod filesystem. Esoteric PEM features such as inter-block comments and block headers are stripped. Certificates are deduplicated. The ordering of certificates within the file is arbitrary, and Kubelet may change the order over time.
`configMap`	`object`	configMap information about the configMap data to project
`downwardAPI`	`object`	downwardAPI information about the downwardAPI data to project
`podCertificate`	`object`	Projects an auto-rotating credential bundle (private key and certificate chain) that the pod can use either as a TLS client or server. Kubelet generates a private key and uses it to send a PodCertificateRequest to the named

Property	Type	Description
		signer. Once the signer approves the request and issues a certificate chain, Kubelet writes the key and certificate chain to the pod filesystem. The pod does not start until certificates have been issued for each podCertificate projected volume source in its spec. Kubelet will begin trying to rotate the certificate at the time indicated by the signer using the PodCertificateRequest.Status.BeginRefreshAt timestamp. Kubelet can write a single file, indicated by the credentialBundlePath field, or separate files, indicated by the keyPath and certificateChainPath fields. The credential bundle is a single file in PEM format. The first PEM entry is the private key (in PKCS#8 format), and the remaining PEM entries are the certificate chain issued by the signer (typically, signers will return their certificate chain in leaf-to-root order). Prefer using the credential bundle format, since your application code can read it atomically. If you use keyPath and certificateChainPath, your application must make two separate file reads. If these coincide with a certificate rotation, it is possible that the private key and leaf certificate you read may not correspond to each other. Your application will need to check for this condition, and re-read until they are consistent. The named signer controls chooses the format of the certificate it issues; consult the signer

Property	Type	Description
		implementation's documentation to learn how to use the certificates it issues.
`secret`	`object`	secret information about the secret data to project
`serviceAccountToken`	`object`	serviceAccountToken is information about the serviceAccountToken data to project

.spec.resources.kubernetesResource.spec.template.spec.volumes[].projected.sources[].clusterTrustBundle

Description

ClusterTrustBundle allows a pod to access the `.spec.trustBundle` field of ClusterTrustBundle objects in an auto-updating file. Alpha, gated by the ClusterTrustBundleProjection feature gate. ClusterTrustBundle objects can either be selected by name, or by the combination of signer name and a label selector. Kubelet performs aggressive normalization of the PEM contents written into the pod filesystem. Esoteric PEM features such as inter-block comments and block headers are stripped. Certificates are deduplicated. The ordering of certificates within the file is arbitrary, and Kubelet may change the order over time.

Type

`object`

Required

`path`

Property	Type	Description
`labelSelector`	`object`	Select all ClusterTrustBundles that match this label selector. Only has effect if signerName is set. Mutually-

Property	Type	Description
		exclusive with name. If unset, interpreted as "match nothing". If set but empty, interpreted as "match everything".
`name`	`string`	Select a single ClusterTrustBundle by object name. Mutually-exclusive with signerName and labelSelector.
`optional`	`boolean`	If true, don't block pod startup if the referenced ClusterTrustBundle(s) aren't available. If using name, then the named ClusterTrustBundle is allowed not to exist. If using signerName, then the combination of signerName and labelSelector is allowed to match zero ClusterTrustBundles.
`path`	`string`	Relative path from the volume root to write the bundle.
`signerName`	`string`	Select all ClusterTrustBundles that match this signer name. Mutually-exclusive with name. The contents of all selected ClusterTrustBundles will be unified and deduplicated.

.spec.resources.kubernetesResource.spec.template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector

Description

Select all ClusterTrustBundles that match this label selector. Only has effect if signerName is set. Mutually-exclusive with name. If unset, interpreted as "match nothing". If set but empty, interpreted as "match everything".

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.resources.kubernetesResource.spec.template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector.matchExpressions

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.volumes[].projected.sources[].clusterTrustBundle.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.resources.kubernetesResource.spec.template.spec.volumes[].projected.sources[].clusterTrustBundle.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.resources.kubernetesResource.spec.template.spec.volumes[].projected.sources[].clusterTrustBundle.labelSelector.matchExpressions[].values[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.volumes[].projected.sources[].clusterTrustBundle.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.resources.kubernetesResource.spec.template.spec.volumes[].projected.sources[].configMap

Description

configMap information about the configMap data to project

Type

`object`

Property	Type	Description
`items`	`array`	items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
`name`	`string`	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`optional`	`boolean`	optional specify whether the ConfigMap or its keys must be defined

.spec.resources.kubernetesResource.spec.template.spec.volumes[].projected.sources[].configMap.items

Description

items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will

error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.volumes[].projected.sources[].configMap.items[]

Description

Maps a string key to a path within a volume.

Type

`object`

Required

`key` `path`

Property	Type	Description
`key`	`string`	key is the key to project.
`mode`	`integer`	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

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

.spec.resources.kubernetesResource.spec.template.spec.volumes[].projected.sources[].downwardAPI

Description

downwardAPI information about the downwardAPI data to project

Type

`object`

Property	Type	Description
`items`	`array`	Items is a list of DownwardAPIVolume file

.spec.resources.kubernetesResource.spec.template.spec.volumes[].projected.sources[].downwardAPI.items

Description

Items is a list of DownwardAPIVolume file

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.volumes[].projected.sources[].downwardAPI.items[]

Description

DownwardAPIVolumeFile represents information to create the file containing the pod field

Type

`object`

Required

`path`

Property	Type	Description
`fieldRef`	`object`	Required: Selects a field of the pod: only annotations, labels, name, namespace and uid are supported.
`mode`	`integer`	Optional: mode bits used to set permissions on this file, must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
`path`	`string`	Required: Path is the relative path name of the file to be created. Must not be absolute or contain the '..' path. Must be utf-8 encoded. The first item of the relative path must not start with '..'
`resourceFieldRef`	`object`	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory,

Property	Type	Description
		requests.cpu and requests.memory) are currently supported.

.spec.resources.kubernetesResource.spec.template.spec.volumes[].projected.sources[].downwardAPI.items[].fieldRef

Description

Required: Selects a field of the pod: only annotations, labels, name, namespace and uid are supported.

Type

`object`

Required

`fieldPath`

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

.spec.resources.kubernetesResource.spec.template.spec.volumes[].projected.sources[].downwardAPI.items[].resourceFieldRef

Description

Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, requests.cpu and requests.memory) are currently supported.

Type

`object`

Required

`resource`

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

.spec.resources.kubernetesResource.spec.template.spec.volumes[].projected.sources[].podCertificate

Description

Projects an auto-rotating credential bundle (private key and certificate chain) that the pod can use either as a TLS client or server. Kubelet generates a private key and uses it to send a PodCertificateRequest to the named signer. Once the signer approves the request and issues a certificate chain, Kubelet writes the key and certificate chain to the pod filesystem. The pod does not start until certificates have been issued for each podCertificate projected volume source in its spec. Kubelet will begin trying to rotate the certificate at the time indicated by the signer using the PodCertificateRequest.Status.BeginRefreshAt timestamp. Kubelet can write a single file, indicated by the credentialBundlePath field, or separate files, indicated by the keyPath and certificateChainPath fields. The credential bundle is a single

file in PEM format. The first PEM entry is the private key (in PKCS#8 format), and the remaining PEM entries are the certificate chain issued by the signer (typically, signers will return their certificate chain in leaf-to-root order). Prefer using the credential bundle format, since your application code can read it atomically. If you use keyPath and certificateChainPath, your application must make two separate file reads. If these coincide with a certificate rotation, it is possible that the private key and leaf certificate you read may not correspond to each other. Your application will need to check for this condition, and re-read until they are consistent. The named signer controls chooses the format of the certificate it issues; consult the signer implementation's documentation to learn how to use the certificates it issues.

Type

`object`

Required

`keyType` `signerName`

Property	Type	Description
`certificateChainPath`	`string`	Write the certificate chain at this path in the projected volume. Most applications should use credentialBundlePath. When using keyPath and certificateChainPath, your application needs to check that the key and leaf certificate are consistent, because it is possible to read the files mid-rotation.
`credentialBundlePath`	`string`	Write the credential bundle at this path in the projected volume. The credential bundle is a single file that contains multiple PEM blocks. The first PEM block is a PRIVATE KEY block, containing a PKCS#8 private key.

Property	Type	Description
		The remaining blocks are CERTIFICATE blocks, containing the issued certificate chain from the signer (leaf and any intermediates). Using credentialBundlePath lets your Pod's application code make a single atomic read that retrieves a consistent key and certificate chain. If you project them to separate files, your application code will need to additionally check that the leaf certificate was issued to the key.
`keyPath`	`string`	Write the key at this path in the projected volume. Most applications should use credentialBundlePath. When using keyPath and certificateChainPath, your application needs to check that the key and leaf certificate are consistent, because it is possible to read the files mid-rotation.
`keyType`	`string`	The type of keypair Kubelet will generate for the pod. Valid values are "RSA3072", "RSA4096", "ECDSAP256", "ECDSAP384", "ECDSAP521", and "ED25519".
`maxExpirationSeconds`	`integer`	maxExpirationSeconds is the maximum lifetime permitted for the certificate.

Property	Type	Description
		Kubelet copies this value verbatim into the PodCertificateRequests it generates for this projection. If omitted, kube-apiserver will set it to 86400(24 hours). kube-apiserver will reject values shorter than 3600 (1 hour). The maximum allowable value is 7862400 (91 days). The signer implementation is then free to issue a certificate with any lifetime *shorter* than MaxExpirationSeconds, but no shorter than 3600 seconds (1 hour). This constraint is enforced by kube-apiserver. `kubernetes.io` signers will never issue certificates with a lifetime longer than 24 hours.
`signerName`	`string`	Kubelet's generated CSRs will be addressed to this signer.
`userAnnotations`	`object`	userAnnotations allow pod authors to pass additional information to the signer implementation. Kubernetes does not restrict or validate this metadata in any way. These values are copied verbatim into the `spec.unverifiedUserAnnotations` field of the PodCertificateRequest objects that Kubelet creates. Entries are subject to the same validation as object metadata annotations, with the addition that all keys must be domain-prefixed. No

Property	Type	Description
		restrictions are placed on values, except an overall size limitation on the entire field. Signers should document the keys and values they support. Signers should deny requests that contain keys they do not recognize.

.spec.resources.kubernetesResource.spec.template.spec.volumes[].projected.sources[].podCertificate.userAnnotations

Description

userAnnotations allow pod authors to pass additional information to the signer implementation. Kubernetes does not restrict or validate this metadata in any way. These values are copied verbatim into the `spec.unverifiedUserAnnotations` field of the PodCertificateRequest objects that Kubelet creates. Entries are subject to the same validation as object metadata annotations, with the addition that all keys must be domain-prefixed. No restrictions are placed on values, except an overall size limitation on the entire field. Signers should document the keys and values they support. Signers should deny requests that contain keys they do not recognize.

Type

`object`

.spec.resources.kubernetesResource.spec.template.spec.volumes[].projected.sources[].secret

Description

secret information about the secret data to project

Type

`object`

Property	Type	Description
`items`	`array`	items if unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
`name`	`string`	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`optional`	`boolean`	optional field specify whether the Secret or its key must be defined

.spec.resources.kubernetesResource.spec.template.spec.volumes[].projected.sources[].secret.items

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.volumes[].projected.sources[].secret.items[]

Description

Maps a string key to a path within a volume.

Type

`object`

Required

`key` `path`

Property	Type	Description
`key`	`string`	key is the key to project.
`mode`	`integer`	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
`path`	`string`	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

.spec.resources.kubernetesResource.spec.template.spec.volumes[].projected.sources[].serviceAccountToken

Description

serviceAccountToken is information about the serviceAccountToken data to project

Type

`object`

Required

`path`

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

.spec.resources.kubernetesResource.spec.template.spec.volumes[].quobyte

Description

quobyte represents a Quobyte mount on the host that shares a pod's lifetime. Deprecated: Quobyte is deprecated and the in-tree quobyte type is no longer supported.

Type

`object`

Required

`registry` `volume`

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

Property	Type	Description
`user`	`string`	user to map volume access to Defaults to serivceaccount user
`volume`	`string`	volume is a string that references an already created Quobyte volume by name.

.spec.resources.kubernetesResource.spec.template.spec.volumes[].rbd

Description

rbd represents a Rados Block Device mount on the host that shares a pod's lifetime. Deprecated: RBD is deprecated and the in-tree rbd type is no longer supported.

Type

`object`

Required

`image` `monitors`

Property	Type	Description
`fsType`	`string`	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#rbd ↗
`image`	`string`	image is the rados image name. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-

Property	Type	Description
		use-it ↗
`keyring`	`string`	keyring is the path to key ring for RBDUser. Default is /etc/ceph/keyring. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it ↗
`monitors`	`array`	monitors is a collection of Ceph monitors. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it ↗
`pool`	`string`	pool is the rados pool name. Default is rbd. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it ↗
`readOnly`	`boolean`	readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it ↗
`secretRef`	`object`	secretRef is name of the authentication secret for RBDUser. If provided overrides keyring. Default is nil. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it ↗

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

.spec.resources.kubernetesResource.spec.template.spec.volumes[].rbd.monitors

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.volumes[].rbd.monitors[]

Type

`string`

.spec.resources.kubernetesResource.spec.template.spec.volumes[].rbd.secretRef

Description

secretRef is name of the authentication secret for RBDUser. If provided overrides keyring. Default is nil. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it

Type

`object`

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

.spec.resources.kubernetesResource.spec.template.spec.volumes[].scaleIO

Description

scaleIO represents a ScaleIO persistent volume attached and mounted on Kubernetes nodes. Deprecated: ScaleIO is deprecated and the in-tree scaleIO type is no longer supported.

Type

`object`

Required

`gateway` `secretRef` `system`

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

Property	Type	Description
`protectionDomain`	`string`	protectionDomain is the name of the ScaleIO Protection Domain for the configured storage.
`readOnly`	`boolean`	readOnly Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
`secretRef`	`object`	secretRef references to the secret for ScaleIO user and other sensitive information. If this is not provided, Login operation will fail.
`sslEnabled`	`boolean`	sslEnabled Flag enable/disable SSL communication with Gateway, default false
`storageMode`	`string`	storageMode indicates whether the storage for a volume should be ThickProvisioned or ThinProvisioned. Default is ThinProvisioned.
`storagePool`	`string`	storagePool is the ScaleIO Storage Pool associated with the protection domain.
`system`	`string`	system is the name of the storage system as configured in ScaleIO.

Property	Type	Description
`volumeName`	`string`	volumeName is the name of a volume already created in the ScaleIO system that is associated with this volume source.

.spec.resources.kubernetesResource.spec.template.spec.volumes[].scaleIO.secretRef

Description

secretRef references to the secret for ScaleIO user and other sensitive information. If this is not provided, Login operation will fail.

Type

`object`

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

.spec.resources.kubernetesResource.spec.template.spec.volumes[].secret

Description

secret represents a secret that should populate this volume. More info: https://kubernetes.io/docs/concepts/storage/volumes#secret

Type

`object`

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

.spec.resources.kubernetesResource.spec.template.spec.volumes[].secret.items

Description

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

Type

`array`

.spec.resources.kubernetesResource.spec.template.spec.volumes[].secret.items[]

Description

Maps a string key to a path within a volume.

Type

`object`

Required

`key` `path`

Property	Type	Description
`key`	`string`	key is the key to project.
`mode`	`integer`	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values

Property	Type	Description
		for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
`path`	`string`	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

.spec.resources.kubernetesResource.spec.template.spec.volumes[].storageos

Description

storageOS represents a StorageOS volume attached and mounted on Kubernetes nodes. Deprecated: StorageOS is deprecated and the in-tree storageos type is no longer supported.

Type

`object`

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

Property	Type	Description
`secretRef`	`object`	secretRef specifies the secret to use for obtaining the StorageOS API credentials. If not specified, default values will be attempted.
`volumeName`	`string`	volumeName is the human-readable name of the StorageOS volume. Volume names are only unique within a namespace.
`volumeNamespace`	`string`	volumeNamespace specifies the scope of the volume within StorageOS. If no namespace is specified then the Pod's namespace will be used. This allows the Kubernetes name scoping to be mirrored within StorageOS for tighter integration. Set VolumeName to any name to override the default behaviour. Set to "default" if you are not using namespaces within StorageOS. Namespaces that do not pre-exist within StorageOS will be created.

.spec.resources.kubernetesResource.spec.template.spec.volumes[].storageos.secretRef

Description

secretRef specifies the secret to use for obtaining the StorageOS API credentials. If not specified, default values will be attempted.

Type

`object`

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

.spec.resources.kubernetesResource.spec.template.spec.volumes[].vsphereVolume

Description

vsphereVolume represents a vSphere volume attached and mounted on kubelets host machine. Deprecated: VsphereVolume is deprecated. All operations for the in-tree vsphereVolume type are redirected to the csi.vsphere.vmware.com CSI driver.

Type

`object`

Required

`volumePath`

Property	Type	Description
`fsType`	`string`	fsType is filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.

Property	Type	Description
`storagePolicyID`	`string`	storagePolicyID is the storage Policy Based Management (SPBM) profile ID associated with the StoragePolicyName.
`storagePolicyName`	`string`	storagePolicyName is the storage Policy Based Management (SPBM) profile name.
`volumePath`	`string`	volumePath is the path that identifies vSphere volume vmdk

.spec.resources.kubernetesResource.spec.template.spec.workloadRef

Description

WorkloadRef provides a reference to the Workload object that this Pod belongs to. This field is used by the scheduler to identify the PodGroup and apply the correct group scheduling policies. The Workload object referenced by this field may not exist at the time the Pod is created. This field is immutable, but a Workload object with the same name may be recreated with different policies. Doing this during pod scheduling may result in the placement not conforming to the expected policies.

Type

`object`

Required

`name` `podGroup`

Property	Type	Description
`name`	`string`	Name defines the name of the Workload object this Pod belongs to. Workload must be in the same namespace as the Pod. If it doesn't match any existing Workload, the Pod will remain unschedulable until a Workload object is created and observed by the kube-scheduler. It must be a DNS subdomain.
`podGroup`	`string`	PodGroup is the name of the PodGroup within the Workload that this Pod belongs to. If it doesn't match any existing PodGroup within the Workload, the Pod will remain unschedulable until the Workload object is recreated and observed by the kube-scheduler. It must be a DNS label.
`podGroupReplicaKey`	`string`	PodGroupReplicaKey specifies the replica key of the PodGroup to which this Pod belongs. It is used to distinguish pods belonging to different replicas of the same pod group. The pod group policy is applied separately to each replica. When set, it must be a DNS label.

.spec.triggers

Type

`array`

.spec.triggers[]

Description

EventListenerTrigger represents a connection between TriggerBinding, Params, and TriggerTemplate; TriggerBinding provides extracted values for TriggerTemplate to then create resources from. TriggerRef can also be provided instead of TriggerBinding, Interceptors and TriggerTemplate

Type

`object`

Property	Type	Description
`bindings`	`array`	
`interceptors`	`array`	
`name`	`string`	
`serviceAccountName`	`string`	ServiceAccountName optionally associates credentials with each trigger; more granular authorization for who is allowed to utilize the associated pipeline vs. defaulting to whatever permissions are associated with the entire EventListener and associated sink facilitates multi-tenant model based scenarios
`template`	`object`	
`triggerRef`	`string`	

.spec.triggers[].bindings

Type

`array`

.spec.triggers[].bindings[]

Type

`object`

Property	Type	Description
`apiversion`	`string`	APIVersion of the binding ref
`kind`	`string`	Kind can only be provided if Ref is also provided. Defaults to TriggerBinding
`name`	`string`	Name is the name of the binding param Mutually exclusive with Ref
`ref`	`string`	Ref is a reference to a TriggerBinding kind. Mutually exclusive with Name
`value`	`string`	Value is the value of the binding param. Can contain JSONPath Has to be pointer since "" is a valid value Required if Name is also specified.

.spec.triggers[].interceptors

Type

`array`

.spec.triggers[].interceptors[]

Description

TriggerInterceptor provides a hook to intercept and pre-process events

Type

`object`

Required

`ref`

Property	Type	Description
`bitbucket`	`object`	BitbucketInterceptor provides a webhook to intercept and pre-process events
`cel`	`object`	CELInterceptor provides a webhook to intercept and pre-process events
`github`	`object`	Deprecated old fields below
`gitlab`	`object`	GitLabInterceptor provides a webhook to intercept and pre-process events
`name`	`string`	Optional name to identify the current interceptor configuration
`params`	`array`	Params are the params to send to the interceptor
`ref`	`object`	Ref refers to the Interceptor to use

Property	Type	Description
`webhook`	`object`	WebhookInterceptor refers to an old style webhook interceptor service

.spec.triggers[].interceptors[].bitbucket

Description

BitbucketInterceptor provides a webhook to intercept and pre-process events

Type

`object`

Property	Type	Description
`eventTypes`	`array`	
`secretRef`	`object`	SecretRef contains the information required to reference a single secret string This is needed because the other secretRef types are not cross-namespace and do not actually contain the "SecretName" field, which allows us to access a single secret value.

.spec.triggers[].interceptors[].bitbucket.eventTypes

Type

`array`

.spec.triggers[].interceptors[].bitbucket.eventTypes[]

Type

`string`

.spec.triggers[].interceptors[].bitbucket.secretRef

Description

SecretRef contains the information required to reference a single secret string This is needed because the other secretRef types are not cross-namespace and do not actually contain the "SecretName" field, which allows us to access a single secret value.

Type

`object`

Property	Type	Description
`secretKey`	`string`	
`secretName`	`string`	

.spec.triggers[].interceptors[].cel

Description

CELInterceptor provides a webhook to intercept and pre-process events

Type

`object`

Property	Type	Description
`filter`	`string`	
`overlays`	`array`	

.spec.triggers[].interceptors[].cel.overlays

Type

`array`

.spec.triggers[].interceptors[].cel.overlays[]

Description

CELOverlay provides a way to modify the request body using DeprecatedCEL expressions

Type

`object`

Property	Type	Description
`expression`	`string`	
`key`	`string`	

.spec.triggers[].interceptors[].github

Description

Deprecated old fields below

Type

`object`

Property	Type	Description
`eventTypes`	`array`	
`secretRef`	`object`	SecretRef contains the information required to reference a single secret string This is needed because the other secretRef types are not cross-namespace and do not actually contain the "SecretName" field, which allows us to access a single secret value.

.spec.triggers[].interceptors[].github.eventTypes

Type

`array`

.spec.triggers[].interceptors[].github.eventTypes[]

Type

`string`

.spec.triggers[].interceptors[].github.secretRef

Description

SecretRef contains the information required to reference a single secret string This is needed because the other secretRef types are not cross-namespace and do not actually contain the "SecretName" field, which allows us to access a single secret value.

Type

`object`

Property	Type	Description
`secretKey`	`string`	
`secretName`	`string`	

.spec.triggers[].interceptors[].gitlab

Description

GitLabInterceptor provides a webhook to intercept and pre-process events

Type

`object`

Property	Type	Description
`eventTypes`	`array`	
`secretRef`	`object`	SecretRef contains the information required to reference a single secret string This is needed because the other secretRef types are not cross-namespace and do not

Property	Type	Description
		actually contain the "SecretName" field, which allows us to access a single secret value.

.spec.triggers[].interceptors[].gitlab.eventTypes

Type

`array`

.spec.triggers[].interceptors[].gitlab.eventTypes[]

Type

`string`

.spec.triggers[].interceptors[].gitlab.secretRef

Description

SecretRef contains the information required to reference a single secret string This is needed because the other secretRef types are not cross-namespace and do not actually contain the "SecretName" field, which allows us to access a single secret value.

Type

`object`

Property	Type	Description
`secretKey`	`string`	
`secretName`	`string`	

.spec.triggers[].interceptors[].params

Description

Params are the params to send to the interceptor

Type

`array`

.spec.triggers[].interceptors[].params[]

Description

InterceptorParams defines a key-value pair that can be passed on an interceptor

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	
`value`		

.spec.triggers[].interceptors[].ref

Description

Ref refers to the Interceptor to use

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent

Property	Type	Description
`kind`	`string`	InterceptorKind indicates the kind of the Interceptor, namespaced or cluster scoped.
`name`	`string`	Name of the referent; More info: http://kubernetes.io/docs/user-guide/identifiers#names ↗

.spec.triggers[].interceptors[].webhook

Description

WebhookInterceptor refers to an old style webhook interceptor service

Type

`object`

Property	Type	Description
`header`	`array`	Header is a group of key-value pairs that can be appended to the interceptor request headers. This allows the interceptor to make decisions specific to an EventListenerTrigger.
`objectRef`	`object`	ObjectRef is a reference to an object that will resolve to a cluster DNS name to use as the EventInterceptor. Either objectRef or url can be specified
`url`	`string`	

.spec.triggers[].interceptors[].webhook.header

Description

Header is a group of key-value pairs that can be appended to the interceptor request headers. This allows the interceptor to make decisions specific to an EventListenerTrigger.

Type

`array`

.spec.triggers[].interceptors[].webhook.header[]

Description

Param declares an ParamValues to use for the parameter called name.

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	
`value`		

.spec.triggers[].interceptors[].webhook.objectRef

Description

ObjectRef is a reference to an object that will resolve to a cluster DNS name to use as the EventInterceptor. Either objectRef or url can be specified

Type

`object`

Property	Type	Description
`apiVersion`	`string`	API version of the referent.
`fieldPath`	`string`	If referring to a piece of an object instead of an entire object, this string should contain a valid JSON/Go field access statement, such as desiredState.manifest.containers[2]. For example, if the object reference is to a container within a pod, this would take on a value like: "spec.containers{name}" (where "name" refers to the name of the container that triggered the event) or if no container name is specified "spec.containers[2]" (container with index 2 in this pod). This syntax is chosen only to have some well-defined way of referencing a part of an object.
`kind`	`string`	Kind of the referent. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`name`	`string`	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
`namespace`	`string`	Namespace of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/ ↗

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

.spec.triggers[].template

Type

`object`

Property	Type	Description
`apiversion`	`string`	
`ref`	`string`	
`spec`	`object`	TriggerTemplateSpec holds the desired state of TriggerTemplate

.spec.triggers[].template.spec

Description

TriggerTemplateSpec holds the desired state of TriggerTemplate

Type

`object`

Property	Type	Description
`params`	`array`	
`resourcetemplates`	`array`	

.spec.triggers[].template.spec.params

Type

`array`

.spec.triggers[].template.spec.params[]

Description

ParamSpec defines an arbitrary named input whose value can be supplied by a `Param`.

Type

`object`

Required

`name`

Property	Type	Description
`default`	`string`	Default is the value a parameter takes if no input value via a Param is supplied.
`description`	`string`	Description is a user-facing description of the parameter that may be used to populate a UI.

Property	Type	Description
`name`	`string`	Name declares the name by which a parameter is referenced.

.spec.triggers[].template.spec.resourcetemplates

Type

`array`

.spec.triggers[].template.spec.resourcetemplates[]

Description

TriggerResourceTemplate describes a resource to create

Type

`object`

.status

Description

EventListenerStatus holds the status of the EventListener

Type

`object`

Required

`configuration`

Property	Type	Description
`address`	`object`	Address is a single Addressable address. If Addresses is present, Address will be ignored by

Property	Type	Description
		clients.
`addresses`	`array`	Addresses is a list of addresses for different protocols (HTTP and HTTPS) If Addresses is present, Address must be ignored by clients.
`annotations`	`object`	Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.
`conditions`	`array`	Conditions the latest available observations of a resource's current state.
`configuration`	`object`	Configuration stores configuration for the EventListener service
`observedGeneration`	`integer`	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

.status.address

Description

Address is a single Addressable address. If Addresses is present, Address will be ignored by clients.

Type

`object`

Property	Type	Description
`CACerts`	`string`	CACerts is the Certification Authority (CA) certificates in PEM format according to https://www.rfc-editor.org/rfc/rfc7468↗.
`hostname`	`string`	
`name`	`string`	Name is the name of the address.
`url`	`string`	

.status.addresses

Description

Addresses is a list of addresses for different protocols (HTTP and HTTPS) If Addresses is present, Address must be ignored by clients.

Type

`array`

.status.addresses[]

Description

Addressable provides a generic mechanism for a custom resource definition to indicate a destination for message delivery. Addressable is the schema for the destination information. This is typically stored in the object's `status`, as this information may be generated by the controller.

Type

`object`

Property	Type	Description
`CACerts`	`string`	CACerts is the Certification Authority (CA) certificates in PEM format according to https://www.rfc-editor.org/rfc/rfc7468↗.
`hostname`	`string`	
`name`	`string`	Name is the name of the address.
`url`	`string`	

.status.annotations

Description

Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

Type

`object`

.status.conditions

Description

Conditions the latest available observations of a resource's current state.

Type

`array`

.status.conditions[]

Description

Condition defines a readiness condition for a Knative resource. See: https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#typical-status-properties

Type

`object`

Required

`status` `type`

Property	Type	Description
`lastTransitionTime`	`string`	LastTransitionTime is the last time the condition transitioned from one status to another. We use VolatileTime in place of metav1.Time to exclude this from creating equality.Semantic differences (all other things held constant).
`message`	`string`	A human readable message indicating details about the transition.
`reason`	`string`	The reason for the condition's last transition.
`severity`	`string`	Severity with which to treat failures of this type of condition. When this is not specified, it defaults to Error.
`status`	`string`	Status of the condition, one of True, False, Unknown.

Property	Type	Description
`type`	`string`	Type of condition.

.status.configuration

Description

Configuration stores configuration for the EventListener service

Type

`object`

Required

`generatedName`

Property	Type	Description
`generatedName`	`string`	GeneratedResourceName is the name given to all resources reconciled by the EventListener

API Endpoints

The following API endpoints are available:

- `/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/eventlisteners`
  - `DELETE`: delete collection of EventListener
  - `GET`: list objects of kind EventListener
  - `POST`: create a new EventListener
- `/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/eventlisteners/{name}`

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

- `/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/eventlisteners/{name}/status`
  - `GET` : read status of the specified EventListener
  - `PATCH` : partially update status of the specified EventListener
  - `PUT` : replace status of the specified EventListener

/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/eventlisteners

HTTP method

`DELETE`

Description

delete collection of EventListener

HTTP responses

HTTP code	Response body
200 - OK	`Status` schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

list objects of kind EventListener

HTTP responses

HTTP code	Response body
200 - OK	`EventListenerList` schema
401 - Unauthorized	Empty

HTTP method

`POST`

Description

create a new EventListener

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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`EventListener` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`EventListener` schema
201 - Created	`EventListener` schema
202 - Accepted	`EventListener` schema
401 - Unauthorized	Empty

/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/eventlisteners/{name}

HTTP method

`DELETE`

Description

delete the specified EventListener

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 EventListener

HTTP responses

HTTP code	Response body
200 - OK	`EventListener` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified EventListener

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

Parameter	Type	Description
		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	`EventListener` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified EventListener

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

Parameter	Type	Description
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`EventListener` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`EventListener` schema
201 - Created	`EventListener` schema
401 - Unauthorized	Empty

/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/eventlisteners/{name}/status

HTTP method

`GET`

Description

read status of the specified EventListener

HTTP responses

HTTP code	Response body
200 - OK	`EventListener` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified EventListener

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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields.

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	`EventListener` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified EventListener

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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object,

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`	`EventListener` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`EventListener` schema
201 - Created	`EventListener` schema
401 - Unauthorized	Empty

TriggerBinding [triggers.tekton.dev/v1beta1]

Description

TriggerBinding defines a mapping of an input event to parameters. This is used to extract information from events to be passed to TriggerTemplates within a Trigger.

Type

`object`

Specification

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

Property	Type	Description
		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`	Spec holds the desired state of the TriggerBinding
`status`	`object`	TriggerBindingStatus defines the observed state of TriggerBinding.

.spec

Description

Spec holds the desired state of the TriggerBinding

Type

`object`

Property	Type	Description
`params`	`array`	Params defines the parameter mapping from the given input event.

.spec.params

Description

Params defines the parameter mapping from the given input event.

Type

`array`

.spec.params[]

Description

Param defines a string value to be used for a ParamSpec with the same name.

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	
`value`	`string`	

.status

Description

TriggerBindingStatus defines the observed state of TriggerBinding.

Type

`object`

API Endpoints

The following API endpoints are available:

- `/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/triggerbindings`
  - `DELETE`: delete collection of TriggerBinding

  - `GET` : list objects of kind TriggerBinding
  - `POST` : create a new TriggerBinding
- `/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/triggerbindings/{name}`
  - `DELETE` : delete the specified TriggerBinding
  - `GET` : read the specified TriggerBinding
  - `PATCH` : partially update the specified TriggerBinding
  - `PUT` : replace the specified TriggerBinding
- `/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/triggerbindings/{name}/status`
  - `GET` : read status of the specified TriggerBinding
  - `PATCH` : partially update status of the specified TriggerBinding
  - `PUT` : replace status of the specified TriggerBinding

/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/triggerbindings

HTTP method

`DELETE`

Description

delete collection of TriggerBinding

HTTP responses

HTTP code	Response body
200 - OK	`Status` schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

list objects of kind TriggerBinding

HTTP responses

HTTP code	Response body
200 - OK	`TriggerBindingList` schema
401 - Unauthorized	Empty

HTTP method

`POST`

Description

create a new TriggerBinding

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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate

Parameter	Type	Description
		fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`TriggerBinding` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`TriggerBinding` schema
201 - Created	`TriggerBinding` schema
202 - Accepted	`TriggerBinding` schema
401 - Unauthorized	Empty

/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/triggerbindings/{name}

HTTP method

`DELETE`

Description

delete the specified TriggerBinding

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will

Parameter	Type	Description
		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 TriggerBinding

HTTP responses

HTTP code	Response body
200 - OK	`TriggerBinding` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified TriggerBinding

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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	`TriggerBinding` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified TriggerBinding

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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`TriggerBinding` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`TriggerBinding` schema

HTTP code	Response body
201 - Created	`TriggerBinding` schema
401 - Unauthorized	Empty

/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/triggerbindings/{name}/status

HTTP method

`GET`

Description

read status of the specified TriggerBinding

HTTP responses

HTTP code	Response body
200 - OK	`TriggerBinding` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified TriggerBinding

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

Parameter	Type	Description
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	`TriggerBinding` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified TriggerBinding

Query parameters

Parameter	Type	Description
`dryRun`	`string`	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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`TriggerBinding` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`TriggerBinding` schema
201 - Created	`TriggerBinding` schema
401 - Unauthorized	Empty

Interceptor [triggers.tekton.dev/v1alpha1]

Description

Interceptor describes a pluggable interceptor including configuration such as the fields it accepts and its deployment address. The type is based on the Validating/MutatingWebhookConfiguration types for configuring AdmissionWebhooks

Type

`object`

Required

`spec`

Specification

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

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

.spec

Description

InterceptorSpec describes the Spec for an Interceptor

Type

`object`

Required

`clientConfig`

Property	Type	Description
`clientConfig`	`object`	ClientConfig describes how a client can communicate with the Interceptor

.spec.clientConfig

Description

ClientConfig describes how a client can communicate with the Interceptor

Type

`object`

Property	Type	Description
`caBundle`	`string`	CaBundle is a PEM encoded CA bundle which will be used to validate the clusterinterceptor server certificate
`service`	`object`	Service is a reference to a Service object where the interceptor is running Mutually exclusive with URL
`url`	`string`	URL is a fully formed URL pointing to the interceptor Mutually exclusive with Service

.spec.clientConfig.service

Description

Service is a reference to a Service object where the interceptor is running Mutually exclusive with URL

Type

`object`

Required

`name` `namespace`

Property	Type	Description
`name`	`string`	Name is the name of the service
`namespace`	`string`	Namespace is the namespace of the service
`path`	`string`	Path is an optional URL path
`port`	`integer`	Port is a valid port number

.status

Description

InterceptorStatus holds the status of the Interceptor

Type

`object`

Property	Type	Description
`address`	`object`	Address is a single Addressable address. If Addresses is present, Address will be ignored by clients.

Property	Type	Description
`addresses`	`array`	Addresses is a list of addresses for different protocols (HTTP and HTTPS) If Addresses is present, Address must be ignored by clients.
`annotations`	`object`	Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.
`conditions`	`array`	Conditions the latest available observations of a resource's current state.
`observedGeneration`	`integer`	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

.status.address

Description

Address is a single Addressable address. If Addresses is present, Address will be ignored by clients.

Type

`object`

Property	Type	Description
`CACerts`	`string`	CACerts is the Certification Authority (CA) certificates in PEM format according to https://www.rfc-editor.org/rfc/rfc7468↗.
`audience`	`string`	Audience is the OIDC audience for this address.
`name`	`string`	Name is the name of the address.
`url`	`string`	

.status.addresses

Description

Addresses is a list of addresses for different protocols (HTTP and HTTPS) If Addresses is present, Address must be ignored by clients.

Type

`array`

.status.addresses[]

Description

Addressable provides a generic mechanism for a custom resource definition to indicate a destination for message delivery. Addressable is the schema for the destination information. This is typically stored in the object's `status`, as this information may be generated by the controller.

Type

`object`

Property	Type	Description
`CACerts`	`string`	CACerts is the Certification Authority (CA) certificates in PEM format according to https://www.rfc-editor.org/rfc/rfc7468.
`audience`	`string`	Audience is the OIDC audience for this address.
`name`	`string`	Name is the name of the address.
`url`	`string`	

.status.annotations

Description

Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

Type

`object`

.status.conditions

Description

Conditions the latest available observations of a resource's current state.

Type

`array`

.status.conditions[]

Description

Condition defines a readiness condition for a Knative resource. See: https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#typical-status-properties

Type

`object`

Required

`status` `type`

Property	Type	Description
`lastTransitionTime`	`string`	LastTransitionTime is the last time the condition transitioned from one status to another. We use VolatileTime in place of metav1.Time to exclude this from creating equality.Semantic differences (all other things held constant).
`message`	`string`	A human readable message indicating details about the transition.
`reason`	`string`	The reason for the condition's last transition.
`severity`	`string`	Severity with which to treat failures of this type of condition. When this is not specified, it defaults to Error.
`status`	`string`	Status of the condition, one of True, False, Unknown.

Property	Type	Description
`type`	`string`	Type of condition.

API Endpoints

The following API endpoints are available:

- `/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/interceptors`
  - `DELETE`: delete collection of Interceptor
  - `GET`: list objects of kind Interceptor
  - `POST`: create a new Interceptor
- `/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/interceptors/{name}`
  - `DELETE`: delete the specified Interceptor
  - `GET`: read the specified Interceptor
  - `PATCH`: partially update the specified Interceptor
  - `PUT`: replace the specified Interceptor
- `/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/interceptors/{name}/status`
  - `GET`: read status of the specified Interceptor
  - `PATCH`: partially update status of the specified Interceptor
  - `PUT`: replace status of the specified Interceptor

/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/interceptors

HTTP method

`DELETE`

Description

delete collection of Interceptor

HTTP responses

HTTP code	Response body
200 - OK	`Status` schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

list objects of kind Interceptor

HTTP responses

HTTP code	Response body
200 - OK	`InterceptorList` schema
401 - Unauthorized	Empty

HTTP method

`POST`

Description

create a new Interceptor

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

Parameter	Type	Description
		processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`Interceptor` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`Interceptor` schema
201 - Created	`Interceptor` schema
202 - Accepted	`Interceptor` schema
401 - Unauthorized	Empty

/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/interceptors/{name}

HTTP method

`DELETE`

Description

delete the specified Interceptor

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 Interceptor

HTTP responses

HTTP code	Response body
200 - OK	`Interceptor` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified Interceptor

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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	`Interceptor` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified Interceptor

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

Parameter	Type	Description
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`Interceptor` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`Interceptor` schema
201 - Created	`Interceptor` schema
401 - Unauthorized	Empty

/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/interceptors/{name}/status

HTTP method

`GET`

Description

read status of the specified Interceptor

HTTP responses

HTTP code	Response body
200 - OK	`Interceptor` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified Interceptor

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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields.

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	`Interceptor` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified Interceptor

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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object,

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`	`Interceptor` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`Interceptor` schema
201 - Created	`Interceptor` schema
401 - Unauthorized	Empty

ClusterTriggerBinding [triggers.tekton.dev/v1beta1]

Description

ClusterTriggerBinding is a TriggerBinding with a cluster scope. ClusterTriggerBindings are used to represent TriggerBindings that should be publicly addressable from any namespace in the cluster.

Type

`object`

Specification

Property	Type	Description
`apiVersion`	`string`	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗
`kind`	`string`	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint

Property	Type	Description
		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`	Spec holds the desired state of the ClusterTriggerBinding from the client
`status`	`object`	TriggerBindingStatus defines the observed state of TriggerBinding.

.spec

Description

Spec holds the desired state of the ClusterTriggerBinding from the client

Type

`object`

Property	Type	Description
`params`	`array`	Params defines the parameter mapping from the given input event.

.spec.params

Description

Params defines the parameter mapping from the given input event.

Type

`array`

.spec.params[]

Description

Param defines a string value to be used for a ParamSpec with the same name.

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	
`value`	`string`	

.status

Description

TriggerBindingStatus defines the observed state of TriggerBinding.

Type

`object`

API Endpoints

The following API endpoints are available:

- `/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/clustertriggerbindings`
  - `DELETE`: delete collection of ClusterTriggerBinding
  - `GET`: list objects of kind ClusterTriggerBinding
  - `POST`: create a new ClusterTriggerBinding
- `/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/clustertriggerbindings/{name}`
  - `DELETE`: delete the specified ClusterTriggerBinding
  - `GET`: read the specified ClusterTriggerBinding
  - `PATCH`: partially update the specified ClusterTriggerBinding
  - `PUT`: replace the specified ClusterTriggerBinding
- `/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/clustertriggerbindings/{name}/status`
  - `GET`: read status of the specified ClusterTriggerBinding
  - `PATCH`: partially update status of the specified ClusterTriggerBinding
  - `PUT`: replace status of the specified ClusterTriggerBinding

/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/clustertriggerbindings

HTTP method

`DELETE`

Description

delete collection of ClusterTriggerBinding

HTTP responses

HTTP code	Response body
200 - OK	`Status` schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method

`GET`

Description

list objects of kind ClusterTriggerBinding

HTTP responses

HTTP code	Response body
200 - OK	`ClusterTriggerBindingList` schema
401 - Unauthorized	Empty

HTTP method

`POST`

Description

create a new ClusterTriggerBinding

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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a

Parameter	Type	Description
		warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`ClusterTriggerBinding` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`ClusterTriggerBinding` schema
201 - Created	`ClusterTriggerBinding` schema
202 - Accepted	`ClusterTriggerBinding` schema
401 - Unauthorized	Empty

/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/clustertriggerbindings/{name}

HTTP method

`DELETE`

Description

delete the specified ClusterTriggerBinding

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 ClusterTriggerBinding

HTTP responses

HTTP code	Response body
200 - OK	`ClusterTriggerBinding` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified ClusterTriggerBinding

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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	`ClusterTriggerBinding` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified ClusterTriggerBinding

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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`ClusterTriggerBinding` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`ClusterTriggerBinding` schema
201 - Created	`ClusterTriggerBinding` schema
401 - Unauthorized	Empty

/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/clustertriggerbindings/{name}/status

HTTP method

`GET`

Description

read status of the specified ClusterTriggerBinding

HTTP responses

HTTP code	Response body
200 - OK	`ClusterTriggerBinding` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified ClusterTriggerBinding

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

Parameter	Type	Description
		processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	`ClusterTriggerBinding` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified ClusterTriggerBinding

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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`ClusterTriggerBinding` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`ClusterTriggerBinding` schema
201 - Created	`ClusterTriggerBinding` schema

HTTP code	Response body
401 - Unauthorized	Empty

ClusterInterceptor [triggers.tekton.dev/v1alpha1]

Description

ClusterInterceptor describes a pluggable interceptor including configuration such as the fields it accepts and its deployment address. The type is based on the Validating/MutatingWebhookConfiguration types for configuring AdmissionWebhooks

Type

`object`

Required

`spec`

Specification

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

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

.spec

Description

ClusterInterceptorSpec describes the Spec for an ClusterInterceptor

Type

`object`

Required

`clientConfig`

Property	Type	Description
`clientConfig`	`object`	ClientConfig describes how a client can communicate with the Interceptor

.spec.clientConfig

Description

ClientConfig describes how a client can communicate with the Interceptor

Type

`object`

Property	Type	Description
`caBundle`	`string`	CaBundle is a PEM encoded CA bundle which will be used to validate the clusterinterceptor server certificate
`service`	`object`	Service is a reference to a Service object where the interceptor is running Mutually exclusive with URL
`url`	`string`	URL is a fully formed URL pointing to the interceptor Mutually exclusive with Service

.spec.clientConfig.service

Description

Service is a reference to a Service object where the interceptor is running Mutually exclusive with URL

Type

`object`

Required

`name` `namespace`

Property	Type	Description
`name`	`string`	Name is the name of the service
`namespace`	`string`	Namespace is the namespace of the service
`path`	`string`	Path is an optional URL path
`port`	`integer`	Port is a valid port number

.status

Description

ClusterInterceptorStatus holds the status of the ClusterInterceptor

Type

`object`

Property	Type	Description
`address`	`object`	Address is a single Addressable address. If Addresses is present, Address will be ignored by clients.

Property	Type	Description
`addresses`	`array`	Addresses is a list of addresses for different protocols (HTTP and HTTPS) If Addresses is present, Address must be ignored by clients.
`annotations`	`object`	Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.
`conditions`	`array`	Conditions the latest available observations of a resource's current state.
`observedGeneration`	`integer`	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

.status.address

Description

Address is a single Addressable address. If Addresses is present, Address will be ignored by clients.

Type

`object`

Property	Type	Description
`CACerts`	`string`	CACerts is the Certification Authority (CA) certificates in PEM format according to https://www.rfc-editor.org/rfc/rfc7468↗.
`audience`	`string`	Audience is the OIDC audience for this address.
`name`	`string`	Name is the name of the address.
`url`	`string`	

.status.addresses

Description

Addresses is a list of addresses for different protocols (HTTP and HTTPS) If Addresses is present, Address must be ignored by clients.

Type

`array`

.status.addresses[]

Description

Addressable provides a generic mechanism for a custom resource definition to indicate a destination for message delivery. Addressable is the schema for the destination information. This is typically stored in the object's `status`, as this information may be generated by the controller.

Type

`object`

Property	Type	Description
`CACerts`	`string`	CACerts is the Certification Authority (CA) certificates in PEM format according to https://www.rfc-editor.org/rfc/rfc7468.
`audience`	`string`	Audience is the OIDC audience for this address.
`name`	`string`	Name is the name of the address.
`url`	`string`	

.status.annotations

Description

Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

Type

`object`

.status.conditions

Description

Conditions the latest available observations of a resource's current state.

Type

`array`

.status.conditions[]

Description

Condition defines a readiness condition for a Knative resource. See: https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#typical-status-properties

Type

`object`

Required

`status` `type`

Property	Type	Description
`lastTransitionTime`	`string`	LastTransitionTime is the last time the condition transitioned from one status to another. We use VolatileTime in place of metav1.Time to exclude this from creating equality.Semantic differences (all other things held constant).
`message`	`string`	A human readable message indicating details about the transition.
`reason`	`string`	The reason for the condition's last transition.
`severity`	`string`	Severity with which to treat failures of this type of condition. When this is not specified, it defaults to Error.
`status`	`string`	Status of the condition, one of True, False, Unknown.

Property	Type	Description
`type`	`string`	Type of condition.

API Endpoints

The following API endpoints are available:

- `/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/clusterinterceptors`
  - `DELETE`: delete collection of ClusterInterceptor
  - `GET`: list objects of kind ClusterInterceptor
  - `POST`: create a new ClusterInterceptor
- `/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/clusterinterceptors/{name}`
  - `DELETE`: delete the specified ClusterInterceptor
  - `GET`: read the specified ClusterInterceptor
  - `PATCH`: partially update the specified ClusterInterceptor
  - `PUT`: replace the specified ClusterInterceptor
- `/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/clusterinterceptors/{name}/status`
  - `GET`: read status of the specified ClusterInterceptor
  - `PATCH`: partially update status of the specified ClusterInterceptor
  - `PUT`: replace status of the specified ClusterInterceptor

/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/clusterinterceptors

HTTP method

`DELETE`

Description

delete collection of ClusterInterceptor

HTTP responses

HTTP code	Response body
200 - OK	`Status` schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

list objects of kind ClusterInterceptor

HTTP responses

HTTP code	Response body
200 - OK	`ClusterInterceptorList` schema
401 - Unauthorized	Empty

HTTP method

`POST`

Description

create a new ClusterInterceptor

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

Parameter	Type	Description
		processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`ClusterInterceptor` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`ClusterInterceptor` schema
201 - Created	`ClusterInterceptor` schema
202 - Accepted	`ClusterInterceptor` schema
401 - Unauthorized	Empty

/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/clusterinterceptors/{name}

HTTP method

`DELETE`

Description

delete the specified ClusterInterceptor

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 ClusterInterceptor

HTTP responses

HTTP code	Response body
200 - OK	`ClusterInterceptor` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified ClusterInterceptor

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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	`ClusterInterceptor` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified ClusterInterceptor

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

Parameter	Type	Description
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`ClusterInterceptor` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`ClusterInterceptor` schema
201 - Created	`ClusterInterceptor` schema
401 - Unauthorized	Empty

/apis/triggers.tekton.dev/v1alpha1/namespaces/{namespace}/clusterinterceptors/{name}/status

HTTP method

`GET`

Description

read status of the specified ClusterInterceptor

HTTP responses

HTTP code	Response body
200 - OK	`ClusterInterceptor` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified ClusterInterceptor

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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields.

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	`ClusterInterceptor` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified ClusterInterceptor

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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object,

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`	`ClusterInterceptor` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`ClusterInterceptor` schema
201 - Created	`ClusterInterceptor` schema
401 - Unauthorized	Empty

ScheduledTrigger [tekton.alaudadevops.io/v1alpha1]

Description

ScheduledTrigger is the Schema for the scheduledtriggers API

Type

`object`

Required

`spec`

Specification

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

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

.spec

Description

spec defines the desired state of ScheduledTrigger

Type

`object`

Required

`triggerTemplate`

Property	Type	Description
`params`	`array`	

Property	Type	Description
`schedule`	`string`	The schedule in Cron format, see https://en.wikipedia.org/wiki/Cron ↗.
`timeZone`	`string`	The time zone name for the given schedule, see https://en.wikipedia.org/wiki/List_of_tz_database_time_zon ↗. If not specified, this will default to the time zone of the kube-controller-manager process. The set of valid time zon names and the time zone offset is loaded from the system-wide time zone database by the API server during CronJob validation and the controller manager during execution. If n system-wide time zone database can be found a bundled version of the database is used instead. If the time zone na becomes invalid during the lifetime of a CronJob or due to a change in host configuration, the controller will stop creatin new new Jobs and will create a system event with the reas UnknownTimeZone. More information can be found in https://kubernetes.io/docs/concepts/workloads/controllers/c jobs/#time-zones ↗
`triggerTemplate`	`object`	

.spec.params

Type

`array`

.spec.params[]

Description

Param defines a string value to be used for a ParamSpec with the same name.

Type

`object`

Required

`name` `value`

Property	Type	Description
`name`	`string`	
`value`	`string`	

.spec.triggerTemplate

Type

`object`

Property	Type	Description
`apiversion`	`string`	
`ref`	`string`	
`spec`	`object`	TriggerTemplateSpec holds the desired state of TriggerTemplate

.spec.triggerTemplate.spec

Description

TriggerTemplateSpec holds the desired state of TriggerTemplate

Type

`object`

Property	Type	Description
`params`	`array`	
`resourcetemplates`	`array`	

.spec.triggerTemplate.spec.params

Type

`array`

.spec.triggerTemplate.spec.params[]

Description

ParamSpec defines an arbitrary named input whose value can be supplied by a `Param`.

Type

`object`

Required

`name`

Property	Type	Description
`default`	`string`	Default is the value a parameter takes if no input value via a Param is supplied.
`description`	`string`	Description is a user-facing description of the parameter that may be used to populate a UI.
`name`	`string`	Name declares the name by which a parameter is referenced.

.spec.triggerTemplate.spec.resourcetemplates

Type

`array`

.spec.triggerTemplate.spec.resourcetemplates[]

Description

TriggerResourceTemplate describes a resource to create

Type

`object`

.status

Description

status defines the observed state of ScheduledTrigger

Type

`object`

Property	Type	Description
`lastScheduleTime`	`string`	Information when was the last time the job was successfully scheduled.

API Endpoints

The following API endpoints are available:

- `/apis/tekton.alaudadevops.io/v1alpha1/namespaces/{namespace}/scheduledtriggers`
  - `DELETE`: delete collection of ScheduledTrigger

  - `GET` : list objects of kind ScheduledTrigger
  - `POST` : create a new ScheduledTrigger
- `/apis/tekton.alaudadevops.io/v1alpha1/namespaces/{namespace}/scheduledtriggers/{name}`
  - `DELETE` : delete the specified ScheduledTrigger
  - `GET` : read the specified ScheduledTrigger
  - `PATCH` : partially update the specified ScheduledTrigger
  - `PUT` : replace the specified ScheduledTrigger
- `/apis/tekton.alaudadevops.io/v1alpha1/namespaces/{namespace}/scheduledtriggers/{name}/status`
  - `GET` : read status of the specified ScheduledTrigger
  - `PATCH` : partially update status of the specified ScheduledTrigger
  - `PUT` : replace status of the specified ScheduledTrigger

/apis/tekton.alaudadevops.io/v1alpha1/namespaces/{namespace}/scheduledtriggers

HTTP method

`DELETE`

Description

delete collection of ScheduledTrigger

HTTP responses

HTTP code	Response body
200 - OK	`Status` schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

list objects of kind ScheduledTrigger

HTTP responses

HTTP code	Response body
200 - OK	`ScheduledTriggerList` schema
401 - Unauthorized	Empty

HTTP method

`POST`

Description

create a new ScheduledTrigger

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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate

Parameter	Type	Description
		fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`ScheduledTrigger` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`ScheduledTrigger` schema
201 - Created	`ScheduledTrigger` schema
202 - Accepted	`ScheduledTrigger` schema
401 - Unauthorized	Empty

/apis/tekton.alaudadevops.io/v1alpha1/namespaces/{namespace}/scheduledtriggers/{name}

HTTP method

`DELETE`

Description

delete the specified ScheduledTrigger

Query parameters

Parameter	Type	Description
`dryRun`	`string`	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will

Parameter	Type	Description
		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 ScheduledTrigger

HTTP responses

HTTP code	Response body
200 - OK	`ScheduledTrigger` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified ScheduledTrigger

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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	`ScheduledTrigger` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified ScheduledTrigger

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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`ScheduledTrigger` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`ScheduledTrigger` schema

HTTP code	Response body
201 - Created	`ScheduledTrigger` schema
401 - Unauthorized	Empty

/apis/tekton.alaudadevops.io/v1alpha1/namespaces/{namespace}/scheduledtriggers/{name}/status

HTTP method

`GET`

Description

read status of the specified ScheduledTrigger

HTTP responses

HTTP code	Response body
200 - OK	`ScheduledTrigger` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified ScheduledTrigger

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

Parameter	Type	Description
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	`ScheduledTrigger` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified ScheduledTrigger

Query parameters

Parameter	Type	Description
`dryRun`	`string`	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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
`body`	`ScheduledTrigger` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`ScheduledTrigger` schema
201 - Created	`ScheduledTrigger` schema
401 - Unauthorized	Empty