Alert APIs

AlertTemplate [alerttemplates.a PrometheusRule [prometheusrules.monitorin

AlertTemplate [alerttemplates.aiops.alauda.io/v1beta1]

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/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/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 contains specification parameters for a template resource

.spec

Description

spec contains specification parameters for a template resource

Type

`object`

Property	Type	Description
`templates`	`array`	templates is a list of alert template

.spec.templates

Description

templates is a list of alert template

Type

`array`

.spec.templates[]

Description

template item

Type

`object`

Required

`compare` `metric` `name` `threshold` `wait`

Property	Type	Description
`annotations`	`object`	rule annotations
`compare`	`string`	compare of a template
`expr`	`string`	query expression
`labels`	`object`	rule labels
`metric`	`object`	metric info for this alert
`name`	`string`	alert range

Property	Type	Description
`metric_name`	`string`	metric name
`notifications`	`array`	notifcations triggered by this alert
`threshold`	`number`	alert threshold
`unit`	`string`	unit for metric
`wait`	`integer`	time of duration

.spec.templates[].annotations

Description

rule annotations

Type

`object`

.spec.templates[].labels

Description

rule labels

Type

`object`

.spec.templates[].metric

Description

metric info for this alert

Type

`object`

Property	Type	Description
`queries`	`array`	queries array for this metric

.spec.templates[].metric.queries

Description

queries array for this metric

Type

`array`

.spec.templates[].metric.queries[]

Type

`object`

Property	Type	Description
`aggregator`	`string`	aggregator functions
`labels`	`array`	labels for alert object

Property	Type	Description
`range`	`integer`	aggregate range

.spec.templates[].metric.queries[].labels

Description

labels for alert object

Type

`array`

.spec.templates[].metric.queries[].labels[]

Type

`object`

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

.spec.templates[].notifications

Description

notifcations triggered by this alert

Type

`array`

.spec.templates[].notifications[]

Type

`object`

Property	Type	Description
`name`	`string`	notification name
`namespace`	`string`	notification namespace
`uuid`	`string`	notification uuid

API Endpoints

The following API endpoints are available:

- `/apis/aiops.alauda.io/v1beta1/alerttemplates`
  - `DELETE`: delete collection of AlertTemplate
  - `GET`: list objects of kind AlertTemplate
  - `POST`: create a new AlertTemplate
- `/apis/aiops.alauda.io/v1beta1/alerttemplates/{name}`
  - `DELETE`: delete the specified AlertTemplate
  - `GET`: read the specified AlertTemplate
  - `PATCH`: partially update the specified AlertTemplate
  - `PUT`: replace the specified AlertTemplate

/apis/aiops.alauda.io/v1beta1/alerttemplates

HTTP method

`DELETE`

Description

delete collection of AlertTemplate

HTTP responses

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

HTTP method

`GET`

Description

list objects of kind AlertTemplate

HTTP responses

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

HTTP method

`POST`

Description

create a new AlertTemplate

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`	`AlertTemplate` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`AlertTemplate` schema
201 - Created	`AlertTemplate` schema

HTTP code	Response body
202 - Accepted	`AlertTemplate` schema
401 - Unauthorized	Empty

/apis/aiops.alauda.io/v1beta1/alerttemplates/{name}

HTTP method

`DELETE`

Description

delete the specified AlertTemplate

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 AlertTemplate

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified AlertTemplate

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	`AlertTemplate` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified AlertTemplate

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`	`AlertTemplate` schema	`application/json` formatted

HTTP responses

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

PrometheusRule [prometheusrules.monitoring.coreos.com/v1]

Description

The `PrometheusRule` custom resource definition (CRD) defines [alerting](https://prometheus.io/docs/prometheus/latest/configuration/alerting_rules/) and [recording](https://prometheus.io/docs/prometheus/latest/configuration/recording_rules/) rules to be evaluated by `Prometheus` or `ThanosRuler` objects. `Prometheus` and `ThanosRuler` objects select `PrometheusRule` objects using label and namespace selectors.

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:

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

.spec

Description

Specification of desired alerting rule definitions for Prometheus.

Type

`object`

Property	Type	Description
`groups`	`array`	Content of Prometheus rule file

.spec.groups

Description

Content of Prometheus rule file

Type

`array`

.spec.groups[]

Description

RuleGroup is a list of sequentially evaluated recording and alerting rules.

Type

`object`

Required

`name`

Property	Type	Description
`interval`	`string`	Interval determines how often rules in the group evaluated.
`labels`	`object`	Labels to add or overwrite before storing the res its rules. The labels defined at the rule level tak precedence. It requires Prometheus >= 3.0.0. The field is ign for Thanos Ruler.

Property	Type	Description
`limit`	`integer`	Limit the number of alerts an alerting rule and s recording rule can produce. Limit is supported s with Prometheus >= 2.31 and Thanos Ruler >=
`name`	`string`	Name of the rule group.
`partial_response_strategy`	`string`	PartialResponseStrategy is only used by Thanc and will be ignored by Prometheus instances. M info: https://github.com/thanos-io/thanos/blob/main/docs/components/rule.md#response ↗
`query_offset`	`string`	Defines the offset the rule evaluation timestamp this particular group by the specified duration in past. It requires Prometheus >= v2.53.0. It is not sup for ThanosRuler.
`rules`	`array`	List of alerting and recording rules.

.spec.groups[].labels

Description

Labels to add or overwrite before storing the result for its rules. The labels defined at the rule level take precedence. It requires Prometheus >= 3.0.0. The field is ignored for Thanos Ruler.

Type

`object`

.spec.groups[].rules

Description

List of alerting and recording rules.

Type

`array`

.spec.groups[].rules[]

Description

Rule describes an alerting or recording rule See Prometheus documentation: [alerting] (https://www.prometheus.io/docs/prometheus/latest/configuration/alerting_rules/) or [recording] (https://www.prometheus.io/docs/prometheus/latest/configuration/recording_rules/#recording-rules) rule

Type

`object`

Required

`expr`

Property	Type	Description
`alert`	`string`	Name of the alert. Must be a valid label value. Only one of `record` and `alert` must be set.
`annotations`	`object`	Annotations to add to each alert. Only valid for alerting rules.

Property	Type	Description
`expr`	`integer|string`	PromQL expression to evaluate.
`for`	`string`	Alerts are considered firing once they have been returned for this long.
`keep_firing_for`	`string`	KeepFiringFor defines how long an alert will continue firing after the condition that triggered it has cleared.
`labels`	`object`	Labels to add or overwrite.
`record`	`string`	Name of the time series to output to. Must be a valid metric name. Only one of `record` and `alert` must be set.

.spec.groups[].rules[].annotations

Description

Annotations to add to each alert. Only valid for alerting rules.

Type

`object`

.spec.groups[].rules[].labels

Description

Labels to add or overwrite.

Type

`object`

API Endpoints

The following API endpoints are available:

- `/apis/monitoring.coreos.com/v1/namespaces/{namespace}/prometheusrules`
  - `DELETE`: delete collection of PrometheusRule
  - `GET`: list objects of kind PrometheusRule
  - `POST`: create a new PrometheusRule
- `/apis/monitoring.coreos.com/v1/namespaces/{namespace}/prometheusrules/{name}`
  - `DELETE`: delete the specified PrometheusRule
  - `GET`: read the specified PrometheusRule
  - `PATCH`: partially update the specified PrometheusRule
  - `PUT`: replace the specified PrometheusRule

/apis/monitoring.coreos.com/v1/namespaces/{namespace}/prometheusrules

HTTP method

`DELETE`

Description

delete collection of PrometheusRule

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 PrometheusRule

HTTP responses

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

HTTP method

`POST`

Description

create a new PrometheusRule

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`	`PrometheusRule` schema	`application/json` formatted

HTTP responses

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

/apis/monitoring.coreos.com/v1/namespaces/{namespace}/prometheusrules/{name}

HTTP method

`DELETE`

Description

delete the specified PrometheusRule

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 PrometheusRule

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified PrometheusRule

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	`PrometheusRule` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified PrometheusRule

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`	`PrometheusRule` schema	`application/json` formatted

HTTP responses

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