

Alert manager

Install Alertmanager

Wget <https://github.com/prometheus/alertmanager/releases/download/v0.25.0/alertmanager-0.25.0.linux-amd64.tar.gz>

```
tar xvf alertmanager-0.25.0.linux-amd64.tar.gz
cd alertmanager-0.25.0.linux-amd64/
./alertmanager
```

```
cat alertmanager.yml
route:
  group_by: ['alertname']
  group_wait: 30s
  group_interval: 5m
  repeat_interval: 1h
  receiver: 'web.hook'
receivers:
- name: 'web.hook'
  webhook_configs:
    - url: 'http://127.0.0.1:5001/'
inhibit_rules:
- source_match:
    severity: 'critical'
  target_match:
    severity: 'warning'
  equal: ['alertname', 'dev', 'instance']
```

Reference:

<https://acloudguru.com/hands-on-labs/installing-prometheus-alertmanager>

Default Config

```
global:
  resolve_timeout: 5m
  http_config:
    follow_redirects: true
    enable_http2: true
  smtp_hello: localhost
  smtp_require_tls: true
  pagerduty_url: https://events.pagerduty.com/v2/enqueue
  opsgenie_api_url: https://api.opsgenie.com/
  wechat_api_url: https://qyapi.weixin.qq.com/cgi-bin/
  victorops_api_url: https://alert.victorops.com/integrations/generic/20131114/alert/
  telegram_api_url: https://api.telegram.org
  webex_api_url: https://webexapis.com/v1/messages
route:
  receiver: web.hook
  group_by:
  - alertname
  continue: false
  group_wait: 30s
  group_interval: 5m
  repeat_interval: 1h
inhibit_rules:
- source_match:
    severity: critical
  target_match:
    severity: warning
  equal:
  - alertname
  - dev
  - instance
receivers:
- name: web.hook
  webhook_configs:
  - send_resolved: true
    http_config:
      follow_redirects: true
      enable_http2: true
      url: http://127.0.0.1:5001/
      max_alerts: 0
templates: []
```

Status

Uptime: 2023-04-20T10:48:00.178Z

Cluster Status

Name: 01GYF4EDXDHKYFD9WCEMJAF5ZS

Status: ready

Peers:

- Name: 01GYF4EDXDHKYFD9WCEMJAF5ZS
- Address: 192.168.207.129:9094

Version Information

Branch: HEAD

BuildDate: 20221222-14:51:36

BuildUser: root@abe866dd5717

GoVersion: go1.19.4

Revision: 258fab7cdd551f2cf251ed0348f0ad7289aee789

Version: 0.25.0

Config

```
global:
  resolve_timeout: 1m
  http_config:
    follow_redirects: true
    enable_http2: true
  smtp_hello: localhost
  smtp_require_tls: true
  slack_api_url: <secret>
  pagerduty_url: https://events.pagerduty.com/v2/enqueue
```

Update alertmanager.yml

```
global:
  resolve_timeout: 1m
  slack_api_url: 'https://hooks.slack.com/services/<ID>'

route:
  receiver: 'slack-notifications'

receivers:
- name: 'slack-notifications'
  slack_configs:
  - send_resolved: true
    channel: '#prometheus'
    icon_url: https://avatars3.githubusercontent.com/u/3380462
    title: |-
      {{{ .Status | toUpper }}}{{{ if eq .Status "firing" }}{{{ .Alerts.Firing | len }}}{{{ end }}} {{{ .CommonLabels.alertname }}} for {{{ .CommonLabels.job }}}
      {{{- if gt (len .CommonLabels) (len .GroupLabels) -}}}
      {{{" "}}}
      {{{- end }}}
    title_link: '{{ template "slack.default.titlelink" . }}'
    text: >-
      {{{ range .Alerts -}}}
      *Alert:* {{{ .Annotations.summary }}}{{{ if .Labels.severity }}} - `{{{ .Labels.severity }}}`{{{ end }}}
      *Description:* {{{ .Annotations.description }}}
      *Details:*
      {{{ range .Labels.SortedPairs }} * {{{ .Name }}}: `{{{ .Value }}}`
      {{{ end }}}
      {{{ end }}}
```

This is a configuration file for Alertmanager, a component of the Prometheus monitoring system that manages alerts sent by Prometheus server. The configuration file defines how alerts are processed and

routed to different receivers, such as email, Slack, PagerDuty, etc.
Here's a breakdown of the different sections and settings in this particular `alertmanager.yml` file:

The `global` section contains global settings for Alertmanager, including:

- `resolve_timeout`: The time to wait before resolving an alert if it has stopped firing (default is 5m).
- `slack_api_url`: The URL for the Slack API webhook that Alertmanager uses to send notifications to Slack. This is specific to the Slack integration and must be configured with your own Slack webhook URL.

The `route` section defines the default receiver for alerts, which in this case is set to `slack-notifications`. This means that any alerts that are not specifically routed to another receiver will be sent to the `slack-notifications` receiver.

The `receivers` section defines the available receivers for alerts, which in this case only includes one receiver named `slack-notifications`. The settings for this receiver include:

- `name`: The name of the receiver.
- `slack_configs`: The Slack-specific configuration for the receiver.

Under the `slack_configs` setting, the following configuration options are defined:

- `send_resolved`: Whether or not to send notifications when an alert has been resolved.
- `channel`: The Slack channel to send the notifications to (e.g. `#prometheus`).
- `icon_url`: The URL for the icon that appears next to the notification in Slack.
- `title`: The title of the notification, which is a template that includes the status of the alert, the name of the alert, and the job associated with the alert.
- `title_link`: A template for the URL to use for the title of the notification, which in this case is using a default template defined elsewhere in the configuration file.
- `text`: The body of the notification, which includes the summary and description of the alert, as well as any additional label values associated with the alert. The `range` function is used to loop over all alerts in the group and create a list of them in the notification body.

Overall, this configuration file is defining how alerts should be sent to Slack and what information should be included in the notification message. It is only one example of how Alertmanager can be configured, and there are many other configuration options and integrations available.

The `title` field in the `slack_configs` section of the `alertmanager.yml` configuration file is a template for the title of the notification that is sent to Slack when an alert is fired. It contains the following elements:

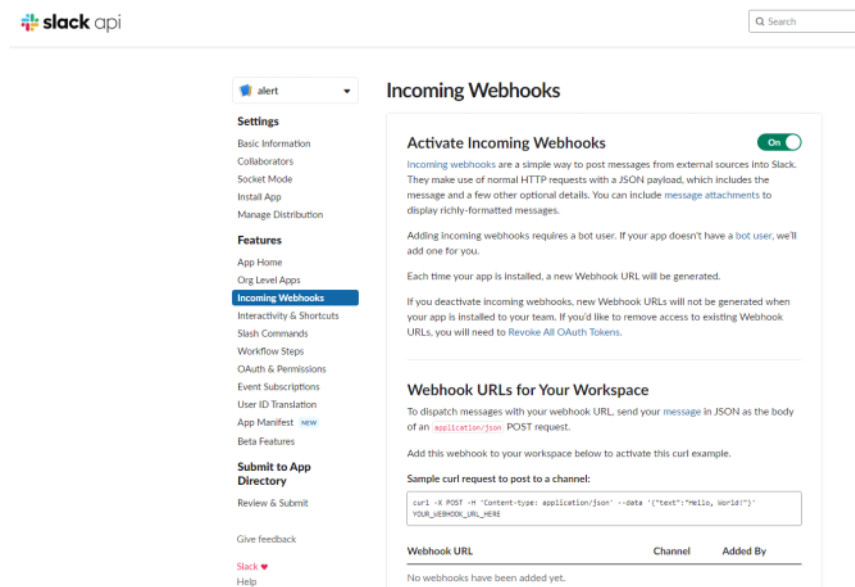
- `| -\n`: This is a YAML block scalar style that indicates a multi-line string literal. The `|` character indicates that the string should be preserved exactly as it is, including line breaks, and the `-` character indicates that any trailing whitespace on each line should be ignored.
- `{{ .Status | toUpper }}`: This is a Go template expression that retrieves the `Status` field of the alert and converts it to uppercase using the `toUpper` function. The `.` character refers to the current context, which is the alert group that triggered the notification.
- `{{ if eq .Status "firing" }}:{{ .Alerts.Firing | len }}{{ end }}`: This is another Go template expression that checks if the `Status` field of the alert is "firing", and if it is, adds a colon and the number of firing alerts to the title. The `Alerts.Firing` field returns a list of all alerts that are currently firing, and the `len` function returns the length of that list.
- `{{ .CommonLabels.alertname }}:{{ .CommonLabels.job }}`: This retrieves the `alertname` and `job` fields from the `CommonLabels` section of the alert group and formats them into a string that describes the alert.
- `{{- if gt (len .CommonLabels) (len .GroupLabels) -}} {{ " " }} {{- end }}`: This is another Go template expression that checks if the number of labels in the `CommonLabels` section is greater than the number of labels in the `GroupLabels` section, and if it is, adds a space to the end of the title. This is used to ensure that the title is aligned properly when different alerts have different label sets.
- `title_link: '{{ template "slack.default.titlelink" . }}'`: This is a reference to another template in the configuration file that specifies the URL to link the title to.
- `text: >- ...`: This is a multi-line string literal that defines the body of the notification, which contains more details about the alerts that have fired. It uses another Go template expression to loop over all alerts in the group and generate a list of their details.

Reference:

- (1) Notification template examples | Prometheus. https://prometheus.io/docs/alerting/latest/notification_examples/.
- (2) Creating Awesome Alertmanager Templates for Slack - Hodovi. <https://hodovi.cc/blog/creating-awesome-alertmanager-templates-for-slack/>.
- (3) Step-by-step guide to setting up Prometheus Alertmanager with Slack <https://grafana.com/blog/2020/02/25/step-by-step-guide-to-setting-up-prometheus-alertmanager-with-slack-pagerduty-and-gmail/>.

https://prometheus.io/docs/alerting/latest/configuration/#slack_config
https://prometheus.io/docs/alerting/latest/notification_examples/

Next, create new webhook and copy like webhook url like tutorial before.



The screenshot shows the Slack API interface for managing incoming webhooks. On the left is a sidebar with navigation links like 'Settings', 'Features', and 'Submit to App Directory'. The main content area is titled 'Incoming Webhooks' and has a toggle switch for 'Activate Incoming Webhooks' which is turned 'On'. Below this, there's explanatory text about how webhooks work. A section titled 'Webhook URLs for Your Workspace' provides a sample curl command to post a message to a channel. At the bottom, there's a table to track added webhooks, but it currently shows 'No webhooks have been added yet.'



alert is requesting permission to access the
devsecops Slack workspace

Where should alert post?

alert requires a channel to post to as an app

random

Cancel

Allow

Add new webhook url into alertmanager.yml

```
global:
  resolve_timeout: 1m
  slack_api_url: 'https://hooks.slack.com/services/<ID>'
```

Use `./-reload` to reload config

```
curl -X POST <IP>:9093/-/reload
```

Go to prometheus dashboard and check runtime Information, we can see alert manager endpoint

Prometheus Alerts Graph Status Help

Runtime Information

Start time	Thu, 20 Apr 2023 09:55:13 GMT
Working directory	/prometheus
Configuration reload	Successful
Last successful configuration reload	2023-04-20T09:55:31Z
WAL corruptions	0
Goroutines	139
GOMAXPROCS	8
GOGC	
GODEBUG	
Storage retention	15d

Build Information

Version	2.34.0
Revision	881111fec4332c33094a6fb2680c71ffc427275
Branch	HEAD
BuildUser	root@121ad7e5487
BuildDate	20220315-15:18:00
GoVersion	go1.17.8

Alertmanagers

Endpoint	http://192.168.207.129:9093/api/v2/alerts
----------	---

CREATE ALERT RULE

First, we go to prometheus config to know simple config.

<https://github.com/istio/istio/blob/master/samples/addons/prometheus.yaml>

```
data:
  allow-snippet-annotations: "false"
  alerting_rules.yml: |
    {}
  alerts: |
```

```
{}
```

This yaml file linked is a configuration file for Prometheus in istio. The data section of the file contains the following key-value pairs:

- allow-snippet-annotations: This setting controls whether Prometheus will allow users to add custom annotations to Prometheus rules. If set to "false", users will only be able to use the pre-defined annotations.
- alerting_rules.yml: This file contains the YAML configuration for Prometheus alerting rules.
- alerts: This file contains the YAML configuration for Prometheus alerts.

The alerting_rules.yml and alerts files are both empty, which means that Prometheus will not be configured to send any alerts by default. If you want to configure Prometheus to send alerts, you will need to add your own rules and alerts to these files.

For more information on configuring Prometheus, please see the Prometheus documentation: <https://prometheus.io/docs/prometheus/latest/configuration/>

Now we update alerting target and rule into prometheus.

```
sudo kubectl -n istio-system get configmap
NAME          DATA  AGE
prometheus    5      9d

sudo kubectl -n istio-system edit configmap prometheus

prometheus.yml: |
global: # This section defines global settings for Prometheus
  evaluation_interval: 1m # Evaluate rules every 1 minute
  scrape_interval: 15s # Scrape targets every 15 seconds
  scrape_timeout: 10s # Timeout for scraping targets
alerting: # This section defines alerting settings for Prometheus
  alertmanagers: # This section specifies the alertmanager instances to send alerts to
    - static_configs: # This section specifies the static list of alertmanager targets
      - targets: http://192.168.207.129:9093 # This is the address of the alertmanager instance

apiVersion: v1 # This is the API version for Kubernetes resources
data: # This section contains the data for the configmap resource
  alerting_rules.yml: | # This is the name of the file that contains the alerting rules
    {
      "groups": [ # This is a list of groups of rules
        {
          "name": "Rules", # This is the name of the group
          "rules": [ # This is a list of rules in the group
            {
              "alert": "InstanceDown", # This is the name of the alert
              "expr": "up == 0", # This is the expression that triggers the alert
              "for": "0m", # This is the duration that the expression must be true before firing the alert
              "annotations": { # This section contains additional information for the alert
                "title": "Instance {{ $Labels.instance }} down", # This is the title of the alert, using label templating
                "description": "{{ $Labels.instance }} of job {{ $Labels.job }} has been down for more than 1 minute." # This is the description of the alert, using label templating
              },
              "labels": { # This section contains additional labels for the alert
                "severity": "critical" # This is a label that indicates the severity of the alert
              }
            },
            {
              "alert": "KubernetesPodClientError", # This is another alert name
              "expr": "istio_requests_total{reporter=\"destination\", response_code=\"403\"} > 10", # This is another expression that triggers the alert, using metric and label filtering
              "labels": { # This section contains additional labels for this alert
                "severity": "warning" # This is another label that indicates the severity of this alert
              },
              "annotations": { # This section contains additional information for this alert
                "summary": "Kubernetes pod Client Error (instance {{ $Labels.instance }})", # This is another title of this alert, using label templating
                "description": "Pod {{ $Labels.instance }} of job {{ $Labels.job }} reported client specific issues" # This is another description of this alert, using label templating
              }
            }
          ]
        }
      ]
    }
  }
}
```

This file is a configuration contains the following settings:

evaluation_interval: This setting controls how often Prometheus will evaluate its rules. The default value is 1 minute.

scrape_interval: This setting controls how often Prometheus will scrape metrics from its targets. The default value is 15 seconds.

scrape_timeout: This setting controls how long Prometheus will wait for a response from a target before giving up. The default value is 10 seconds.

alertmanagers: This section configures Prometheus to send alerts to an Alertmanager. The Alertmanager is a separate service that is responsible for handling alerts and notifying users.

It contains the following rules:

InstanceDown: This rule alerts if an instance has been down for more than 1 minute.

KubernetesPodClientError: This rule alerts if a Kubernetes pod has reported more than 10 client errors.

Each rule has the following settings:

alert: This is the name of the alert.

expr: This is the expression that Prometheus will use to evaluate the rule.

for: This is the duration for which Prometheus will keep an alert in the firing state.

annotations: This is a map of annotations that will be added to alerts that fire.

labels: This is a map of labels that will be added to alerts that fire.

we can add larger rule based on rule file below and using converter yaml to json:

<https://github.com/samber/awesome-prometheus-alerts/blob/master/dist/rules/istio/embedded-exporter.yml>

After update config, we restart pod prometheus:

```
sudo kubectl -n istio-system describe pod -l=app=prometheus
sudo kubectl -n istio-system delete pod -l=app=prometheus
sudo kubectl -n istio-system get pod -l=app=prometheus
sudo kubectl -n istio-system logs prometheus-7cc75b7c8c-nmbmw --container prometheus-server
```

Now, we go to prometheus to check rule again

Rules				
Rules			45.886s ago	0.520ms
Rule	State	Error	Last Evaluation	Evaluation Time
alert: InstanceDown expr: up == 0 labels: severity: critical annotations: description: {{ \$labels.instance }} of job {{ \$labels.job }} has been down for more than 1 minute. title: Instance {{ \$labels.instance }} down	OK		45.888s ago	0.354ms
alert: KubernetesPodClientError expr: info_requests_total{reporter="destination",response_code!="403"} > 10 labels: severity: warning annotations: description: Pod {{ \$labels.instance }} of job {{ \$labels.job }} reported client specific issues summary: Kubernetes pod Client Error (instance {{ \$labels.instance }})	OK		45.888s ago	0.155ms

Prometheus Alerts Graph Status Help	
Runtime Information	
Start time	Mon, 17 Apr 2023 16:11:15 GMT
Working directory	/prometheus
Configuration reload	Successful
Last successful configuration reload	2023-04-17T16:11:15Z
WAL corruptions	0
Goroutines	139
GOMAXPROCS	8
GOGC	
GODEBUG	
Storage retention	15d
Build Information	
Version	2.34.0
Revision	881111fec4332c33094a6fb2680c71fffc427275
Branch	HEAD
BuildUser	root@121ad7ea5487
BuildDate	20220315-15:18:00
GoVersion	go1.17.8
Alertmanagers	
Endpoint	http://192.168.207.129:9093/api/v2/alerts

Disable peer authentication to run test rule.

```
sudo kubectl edit pa -n istio-system
spec:
  mtls:
    mode: DISABLE
```

Create nginx service:

```
kubectl -n prod run nginx --image nginx
kubectl -n prod expose pod nginx --port 80
kubectl -n prod get svc
```

Use curl to test connection:

```
curl <IP nginx svc>:80
or
kubectl -n prod exec -it nginx -- curl 10.32.0.28:80
```

Next, we go to nginx and delete index.html.

```
kubectl -n prod exec -it nginx -- bash
rm /usr/share/nginx/html/index.html
```

We check the connection again by using the following bash script and get a 403 error.
while true; do curl -s 10.101.73.244:80; sleep 1; done

Now let's go back to the dashboard to check if the rule is working

The Prometheus Alerts dashboard shows a list of alerts. The 'KubernetesPodClientError' alert is active and firing. The alert details are as follows:

```
name: KubernetesPodClientError
expr: |istio_requests_total(reporter="destination",response_code="403") > 10
labels:
  severity: warning
annotations:
  description: Pod {{ $labels.instance }} of job {{ $labels.job }} reported client specific issues
  summary: Kubernetes pod Client Error (instance {{ $labels.instance }})
```

Labels	State	Active Since	Value
alertname:KubernetesPodClientError, connection_security_policy:none, destination.canonical_revision:latest, destination.canonical.service:nginx, destination.cluster:Kubernetes, destination.principal:unknown, destination.service:nginx.prod.svc.cluster.local, destination.service.name:nginx, destination.service.namespace:prod, destination.workload:nginx, destination.workload.namespace:prod, instance:10.32.0.28:15020, job:kubernetes-pods, namespace:prod, pod:nginx, reporter:destination, request.protocol:http, response.code=403, response.flags:-, run:nginx, security.istio.io.thMode:istio, service.istio.io.canonical.name:nginx, service.istio.io.canonical.revision:latest, severity:warning, source.app:unknown, source.canonical.revision:latest, source.canonical.service:unknown, source.cluster:unknown, source.principal:unknown, source.version:unknown, source.workload:unknown, source.workload.namespace:unknown	FIRING	2023-04-18T12:53:40.180784161Z	93

we can click on expr to execute the query

The Prometheus query execution interface shows the query: `|istio_requests_total{ reporter="destination", response_code="403" } > 10`. The results are as follows:

Table	Graph
Evaluation time: []	
istio_requests_total{connection_security_policy="none", destination.canonical_revision="latest", destination.canonical.service="nginx", destination.cluster="Kubernetes", destination.principal="unknown", destination.service="nginx.prod.svc.cluster.local", destination.service.name="nginx", destination.service.namespace="prod", destination.workload="nginx", destination.workload.namespace="prod", instance="10.32.0.28:15020", job="kubernetes-pods", namespace="prod", pod="nginx", reporter="destination", request.protocol="http", response.code="403", response.flags="-", run="nginx", security.istio.io.thMode="istio", service.istio.io.canonical.name="nginx", service.istio.io.canonical.revision="latest", source.app="unknown", source.canonical.revision="latest", source.canonical.service="unknown", source.cluster="unknown", source.principal="unknown", source.version="unknown", source.workload="unknown", source.workload.namespace="unknown"} 93	

and alert is also sent to slack

The Slack alert message is as follows:

```
alert APP: 7:52 PM
[FIRING:1] KubernetesPodClientError for kubernetes-pods
Alert: Kubernetes pod Client Error (instance 10.32.0.28:15020) - warning
Description: Pod 10.32.0.28:15020 of job kubernetes-pods reported client specific issues
Details:
  • alertname: KubernetesPodClientError
  • connection_security_policy: mutual_tls
Show more
```

similar to alertmanager, we can select silence.

Filter
Group
Receiver: All
Silenced
Inhibited

+
Silence

Custom matcher, e.g. `env="production"`

+ Expand all groups

- Not grouped 1 alert

2023-04-18T12:53:40.180Z + Info Source Silence Link

```

alertname="KubernetesPodClientError" + connection_security_policy="none" + destination_canonical_revision="latest" +
destination_canonical_service="nginx" + destination_cluster="Kubernetes" + destination_principal="unknown" +
destination_service="nginx.prod.svc.cluster.local" + destination_service_name="nginx" + destination_service_namespace="prod" +
destination_workload="nginx" + destination_workload_namespace="prod" + instance="10.32.0.28:15020" + job="kubernetes-pods" +
namespace="prod" + pod="nginx" + reporter="destination" + request_protocol="http" + response_code="403" + response_flags="-" +
run="nginx" + security_istio_io_tlsMode="istio" + service_istio_io_canonical_name="nginx" + service_istio_io_canonical_revision="latest" +
severity="warning" + source_app="unknown" + source_canonical_revision="latest" + source_canonical_service="unknown" +
source_cluster="unknown" + source_principal="unknown" + source_version="unknown" + source_workload="unknown" +
source_workload_namespace="unknown" +

```

We can enter the creator and comment information to assign the person in charge of handling this case

Silence

Edit Expire

ID e1733ea3-f0ea-4528-ad50-bbc82ba44f2e

Starts at 2023-04-18T13:03:49.351Z

Ends at 2023-04-18T15:02:58.353Z

Updated at 2023-04-18T13:03:49.351Z

Created by some one

Comment this issue have been assign

State active

Matchers

```

alertname=KubernetesPodClientError + connection_security_policy=none + destination_canonical_revision=latest
destination_canonical_service=nginx + destination_cluster=Kubernetes + destination_principal=unknown
destination_service=nginx.prod.svc.cluster.local + destination_service_name=nginx + destination_service_namespace=prod
destination_workload=nginx + destination_workload_namespace=prod + instance=10.32.0.28:15020 + job=kubernetes-pods
namespace=prod + pod=nginx + reporter=destination + request_protocol=http + response_code=403 + response_flags=-
run=nginx + security_istio_io_tlsMode=istio + service_istio_io_canonical_name=nginx + service_istio_io_canonical_revision=latest
severity=warning + source_app=unknown + source_canonical_revision=latest + source_canonical_service=unknown
source_cluster=unknown + source_principal=unknown + source_version=unknown + source_workload=unknown
source_workload_namespace=unknown

```

Affected alerts: 1

2023-04-18T12:53:40.180Z + Info Source

similarly we continue to test the rule instance down by deleting pod nginx:

```
kubectl -n prod delete pod nginx
```

Go to dashboard to run query

Prometheus Alerts Graph Status Help

☒ Use local time ☒ Enable query history ☒ Enable autocomplete ☒ Enable highlighting ☒ Enable linter

up == 0

Table Graph

Evaluation time

up{instance="10.32.0.28:15020", job="kubernetes-pods", namespace="prod", pod="nginx", run="nginx", security_istio_io_tlsMode="istio", service_istio_io_canonical_name="nginx", service_istio_io_canonical_revision="latest"}

And we receive slack alert

alert APP 8:36 PM

[FIRING:1] InstanceDown for kubernetes-pods

Alert: - critical

Description: 10.32.0.28:15020 of job kubernetes-pods has been down for more than 1 minute.

Details:

- alertname: InstanceDown
- instance: 10.32.0.28:15020

Show more