Demo Project:
Configure Alerting for our Application

Project Overview

This guide provides a step-by-step approach to configuring alerting for our application. We will set up monitoring to notify us when CPU usage exceeds 50% or if a pod cannot start. The process includes configuring alert rules in Prometheus, setting up Alertmanager with email notifications, and testing alerts.

Step 1: Configure Alert Rules in Prometheus Server

Refer to:
https://docs.redhat.com/en/documentation/openshift_container_platform/4.18/html/monitoring_apis/prometheusrule-monitoring-coreos-com-v1

1. **Create and add Alert Rules** `alert-rules.yaml` **:**

```
apiVersion: monitoring.coreos.com/v1
kind: PrometheusRule
metadata:
  name: main-rules
  namespace: monitoring
  labels:
    app: kube-prometheus-stack
    release: monitoring
spec:
  groups:
  - name: main.rules
    rules:
    - alert: HostHighCpuLoad
      expr: 100 - (avg by(instance) (rate(node_cpu_seconds_total{mode="idle"}[2m])) * 100) > 50
      for: 2m
      labels:
        severity: warning
        namespace: monitoring
      annotations:
        description: "CPU load on host is over 50%\n Value = {{ $value }}\n Instance = {{ $labels.instance }}\n"
        summary: "Host CPU load high"
    - alert: KubernetesPodCrashLooping
      expr: kube_pod_container_status_restarts_total > 5
      for: 0m
      labels:
        severity: critical
        namespace: monitoring
      annotations:
        description: "Pod {{ $labels.pod }} is crash looping\n Value = {{ $value }}"
        summary: "Kubernetes pod crash looping"
```

2. **Apply Alert Rules:**
   - `kubectl apply -f alert-rules.yaml`

   Verify:
   - `kubectl get PrometheusRule -n monitoring`

     Example Output: `mail-rules` should appear in the list.

3. **Troubleshoot Alert Rules Loading**
   - Check if Prometheus loaded the rules correctly:
     `kubectl get pod -n monitoring`

     Expected output:

     ```
     NAME                                                     READY   STATUS    RESTARTS   AGE
     alertmanager-monitoring-kube-prometheus-alertmanager-0   2/2     Running   0          4h23m
     monitoring-grafana-c6f9bf774-x8nxk                       3/3     Running   0          4h23m
     monitoring-kube-prometheus-operator-77986bdf66-wgrsd     1/1     Running   0          4h23m
     monitoring-kube-state-metrics-7f6cdff9-wdwfd             1/1     Running   0          4h23m
     monitoring-prometheus-node-exporter-dr2qg                1/1     Running   0          4h23m
     monitoring-prometheus-node-exporter-g9tjs                1/1     Running   0          4h23m
     prometheus-monitoring-kube-prometheus-prometheus-0       2/2     Running   0          4h23m
     ```

   - **Check Prometheus Logs**
     - To verify if the alert rules were loaded correctly:

     `kubectl logs prometheus-monitoring-kube-prometheus-prometheus-0 -n monitoring -c config-reloader`

     - If the output contains `msg="Reload triggered"`, the configuration was loaded successfully.

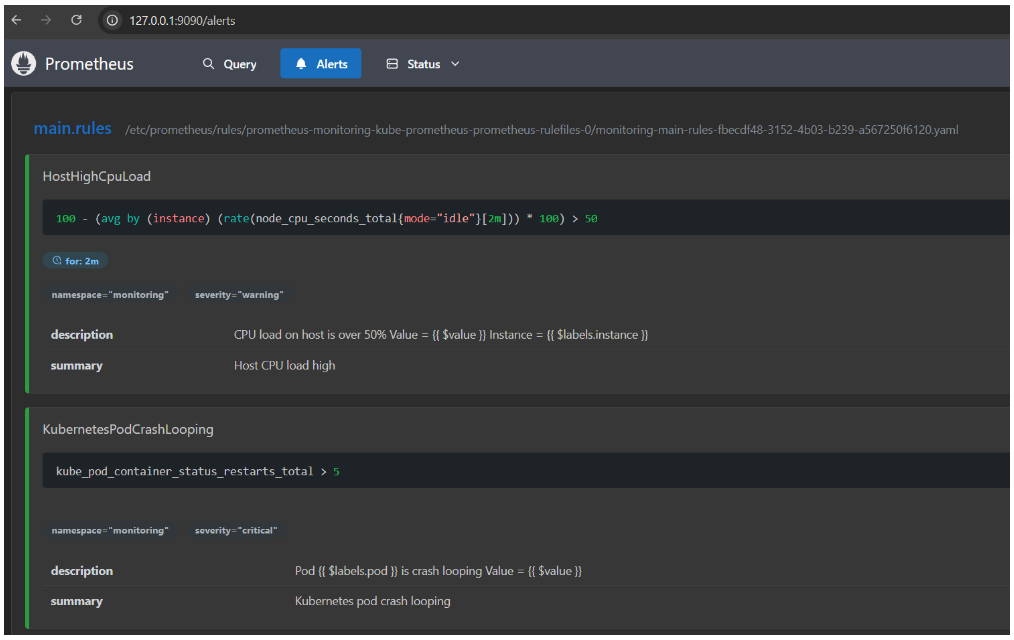
     ```
     level=info ts=2025-03-14T22:38:32.541049437Z caller=reloader.go:548 msg="Reload triggered" cfg_in
     =/etc/prometheus/config/prometheus.yaml.gz cfg_out=/etc/prometheus/config_out/prometheus.env.ya
     ml cfg_dirs= watched_dirs=/etc/prometheus/rules/prometheus-monitoring-kube-prometheus-promethe
     us-rulefiles-0
     ```

   - **Check Prometheus' main logs for further confirmation:**
     `kubectl logs prometheus-monitoring-kube-prometheus-prometheus-0 -n monitoring -c prometheus`
     - If the output includes `Completed loading of configuration file`, Prometheus successfully applied the alert rules.

       ```
       time=2025-03-14T22:38:32.540Z level=INFO source=main.go:1486 msg="Completed loading of con
       figuration file" db_storage=1.377µs remote_storage=1.7µs web_handler=471ns query_engine=1.219µs
       scrape=2.647885ms scrape_sd=70.23µs notify=193.973µs notify_sd=7.681µs rules=55.342609ms tr
       acing=6.068µs filename=/etc/prometheus/config_out/prometheus.env.yaml totalDuration=63.78946
       ms
       ```

- **Check Prometheus UI:**

  Example:



127.0.0.1:9090/alerts

Prometheus · Query · Alerts · Status

main.rules /etc/prometheus/rules/prometheus-monitoring-kube-prometheus-prometheus-rulefiles-0/monitoring-main-rules-fbecdf48-3152-4b03-b239-a567250f6120.yaml

HostHighCpuLoad

```
100 - (avg by (instance) (rate(node_cpu_seconds_total{mode="idle"}[2m])) * 100) > 50
```

for: 2m

namespace="monitoring" severity="warning"

description	CPU load on host is over 50% Value = {{ $value }} Instance = {{ $labels.instance }}
summary	Host CPU load high

KubernetesPodCrashLooping

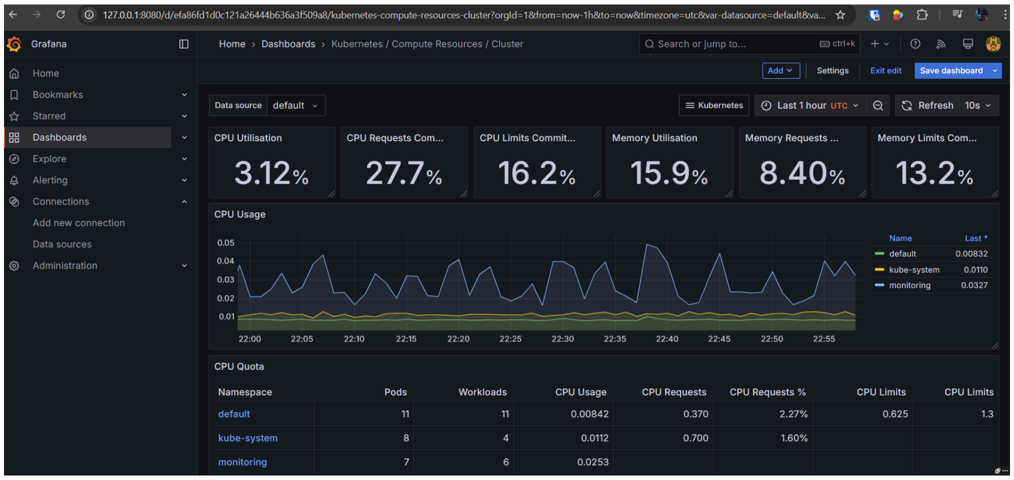
```
kube_pod_container_status_restarts_total > 5
```

namespace="monitoring" severity="critical"

description	Pod {{ $labels.pod }} is crash looping Value = {{ $value }}
summary	Kubernetes pod crash looping

Step 2: Test Alert Rule

1. **Navigate to Kubernetes / Compute Resources / Cluster:**



127.0.0.1:8080/d/efa86fd1d0c121a26444b636a3f509a8/kubernetes-compute-resources-cluster?orgId=1&from=now-1h&to=now&timezone=utc&var-datasource=default&va...

Home › Dashboards › Kubernetes / Compute Resources / Cluster

Data source: default · Last 1 hour UTC · Refresh 10s

CPU Utilisation	CPU Requests Com...	CPU Limits Commit...	Memory Utilisation	Memory Requests ...	Memory Limits Com...
3.12%	27.7%	16.2%	15.9%	8.40%	13.2%

Name	Last *
default	0.00832
kube-system	0.0110
monitoring	0.0327

CPU Quota

Namespace	Pods	Workloads	CPU Usage	CPU Requests	CPU Requests %	CPU Limits	CPU Limits
default	11	11	0.00842	0.370	2.27%	0.625	1.3
kube-system	8	4	0.0112	0.700	1.60%		
monitoring	7	6	0.0253				

2. **Simulate a CPU Load**

   - Find `cpustress` container in Docker hub: https://hub.docker.com/r/containerstack/cpustress
   - Deploy stress pod:

     `kubectl run cpu-test --image=containerstack/cpustress -- --cpu 4 --timeout 60s --metrics-brief`

Note: there is an extra "`--`" since what comes after takes as options or parameters for the application inside the container.

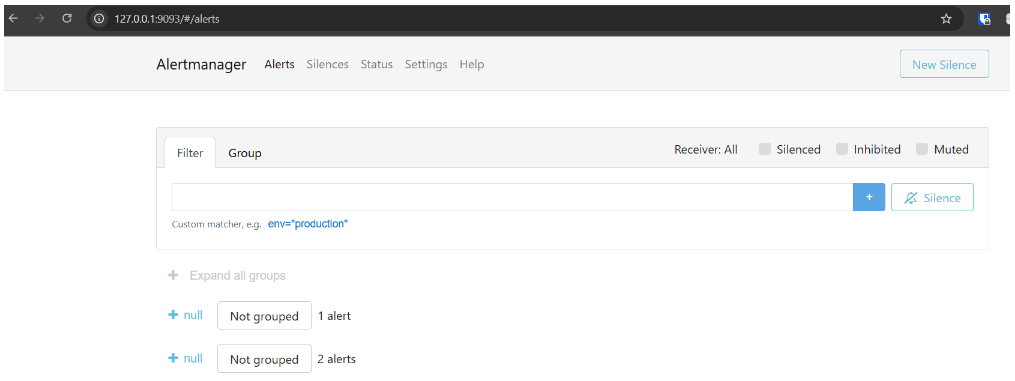
- Confirm pod created: `kubectl get pod`
- (If needed) Delete test pod: `kubectl delete pod cpu-test`

Step 3: Access Alert manager Dashboard

1. **Access Alert manager UI:**

```
kubectl port-forward -n monitoring svc/monitoring-kube-prometheus-alertmanager 9093:9093 &
```

2. **Access Alert manager:**
   - Open a browser and go to: `127.0.0.1:9093`



Step 4: Configure Email Notification

1. **Create:** `alert-manager-configuration.yaml`

```
apiVersion: monitoring.coreos.com/v1beta1
kind: AlertmanagerConfig
metadata:
  name: main-rules-alert-config
  namespace: monitoring
spec:
  route:
    receiver: 'email'
    repeatInterval: 30m
    routes:
    - matchers:
      - name: alertname
        value: HostHighCpuLoad
    - matchers:
      - name: alertname
        value: KubernetesPodCrashLooping
    repeatInterval: 10m
  receivers:
  - name: 'email'
    emailConfigs:
    - to: 'eduardobautista.devops@gmail.com'
      from: 'eduardobautista.devops@gmail.com'
      smarthost: 'smtp.gmail.com:587'
      authUsername: 'eduardobautista.devops@gmail.com'
      authIdentity: 'eduardobautista.devops@gmail.com'
      authPassword:
        name: gmail-auth
        key: password
```

2. **Create Secret for Email Authentication:**

```
apiVersion: v1
kind: Secret
type: Opaque
metadata:
  name: gmail-auth
  namespace: monitoring
data:
  password: base64-encoded-value-of-your-password
```

- **Generate Password (Base64 Encoded)**
    1. Generate app password from: https://myaccount.google.com/u/1/apppassword
    2. Encode the password: `echo -n "your-app-password" | base64`
    3. paste in password

3. **Apply the configurations:**
    - `kubectl apply -f email-secret.yaml`
    - `kubectl apply -f` **`alert-manager-configuration.yaml`**

4. **Verify:**
    - `kubectl get alertmanagerconfig -n monitoring`
    - `kubectl get pod -n monitoring`

    Example output:

```
alertmanager-monitoring-kube-prometheus-alertmanager-0   2/2     Running   0          6h19m
monitoring-grafana-c6f9bf774-x8nxk                       3/3     Running   0          6h19m
monitoring-kube-prometheus-operator-77986bdf66-wgrsd     1/1     Running   0          6h19m
monitoring-kube-state-metrics-7f6cdff9-wdwfd             1/1     Running   0          6h19m
monitoring-prometheus-node-exporter-dr2qg                1/1     Running   0          6h19m
monitoring-prometheus-node-exporter-g9tjs                1/1     Running   0          6h19m
prometheus-monitoring-kube-prometheus-prometheus-0       2/2     Running   0          6h19m
```

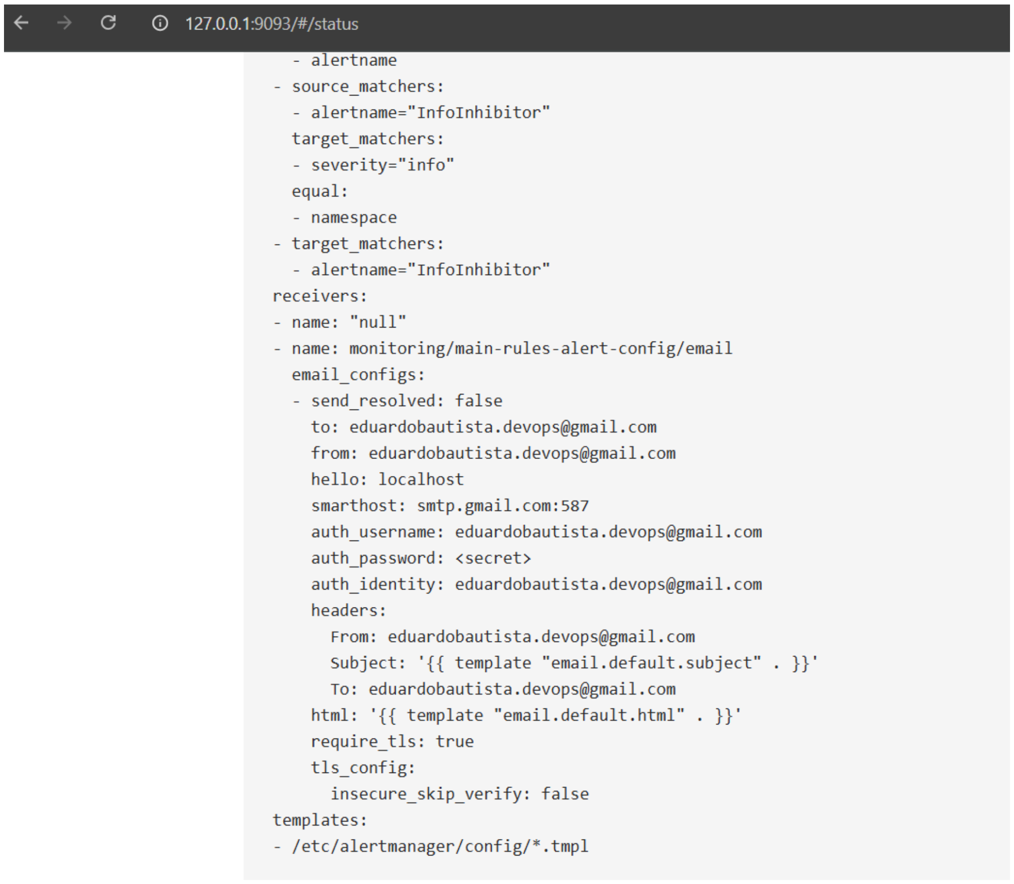
5. **Check Alertmanager Logs:**

- `kubectl logs alertmanager-monitoring-kube-prometheus-alertmanager-0 -n monitoring -c config-reloader`

If successful, logs should contain:

```
level=info ts=2025-03-15T00:34:30.552934557Z caller=reloader.go:548 msg="Reload triggered" cfg_in=/et
c/alertmanager/config/alertmanager.yaml.gz cfg_out=/etc/alertmanager/config_out/alertmanager.env.yaml c
fg_dirs= watched_dirs=/etc/alertmanager/config
```

- In Alert manager UI observe that the configuration was also added:



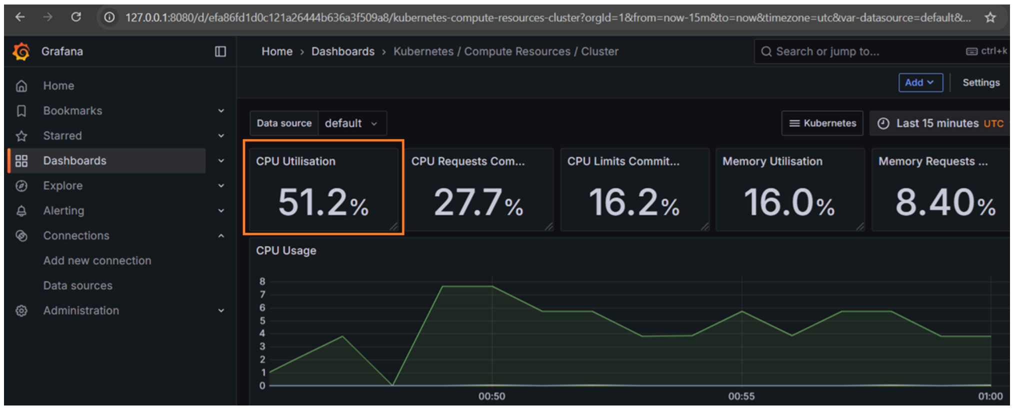
```
    - alertname
  - source_matchers:
    - alertname="InfoInhibitor"
    target_matchers:
    - severity="info"
    equal:
    - namespace
  - target_matchers:
    - alertname="InfoInhibitor"
receivers:
- name: "null"
- name: monitoring/main-rules-alert-config/email
  email_configs:
  - send_resolved: false
    to: eduardobautista.devops@gmail.com
    from: eduardobautista.devops@gmail.com
    hello: localhost
    smarthost: smtp.gmail.com:587
    auth_username: eduardobautista.devops@gmail.com
    auth_password: <secret>
    auth_identity: eduardobautista.devops@gmail.com
    headers:
      From: eduardobautista.devops@gmail.com
      Subject: '{{ template "email.default.subject" . }}'
      To: eduardobautista.devops@gmail.com
    html: '{{ template "email.default.html" . }}'
    require_tls: true
    tls_config:
      insecure_skip_verify: false
templates:
- /etc/alertmanager/config/*.tmpl
```

Step 5: Triggeting the alerts

1. **Trigger CPU Stress**

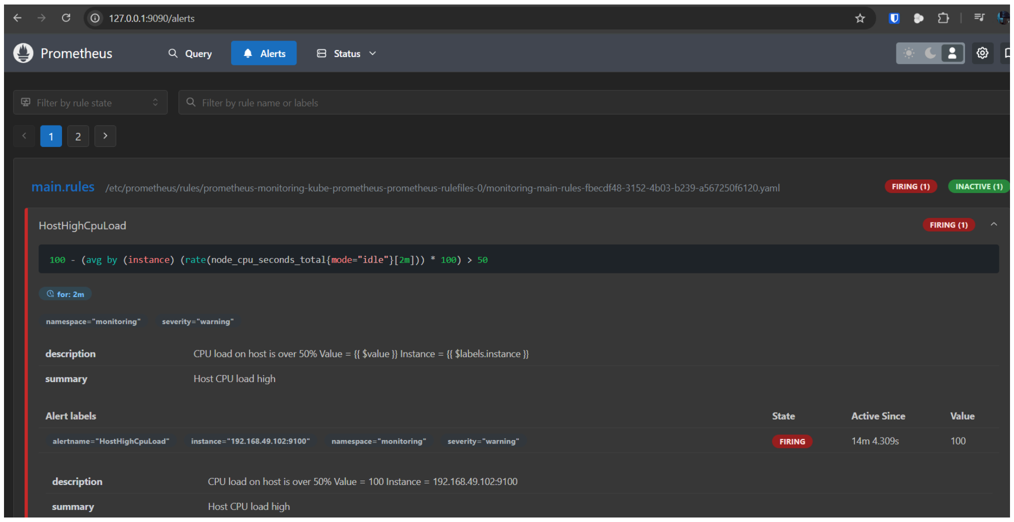
```
kubectl delete pod cpu-test; kubectl run cpu-test --image=containerstack/cpustress -- --cpu 4 --timeout 60s --
metrics-brief
```

2. **Observe spike in Grafana and Prometheus:**


127.0.0.1:8080/d/efa86fd1d0c121a26444b636a3f509a8/kubernetes-compute-resources-cluster?orgId=1&from=now-15m&to=now&timezone=utc&var-datasource=default&...

Data source: default

CPU Utilisation	CPU Requests Com...	CPU Limits Commit...	Memory Utilisation	Memory Requests ...
51.2%	27.7%	16.2%	16.0%	8.40%


127.0.0.1:9090/alerts

main.rules /etc/prometheus/rules/prometheus-monitoring-kube-prometheus-prometheus-rulefiles-0/monitoring-main-rules-fbecdf48-3152-4b03-b239-a567250f6120.yaml FIRING (1) INACTIVE (1)

HostHighCpuLoad FIRING (1)

```
100 - (avg by (instance) (rate(node_cpu_seconds_total{mode="idle"}[2m])) * 100) > 50
```

for: 2m

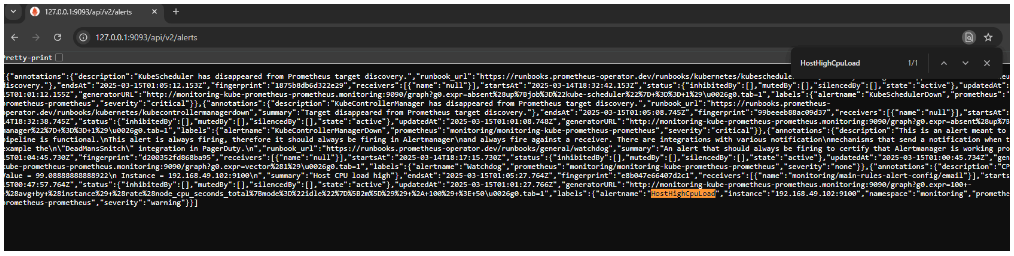
namespace="monitoring" severity="warning"

description	CPU load on host is over 50% Value = {{ $value }} Instance = {{ $labels.instance }}
summary	Host CPU load high

Alert labels	State	Active Since	Value
alertname="HostHighCpuLoad" instance="192.168.49.102:9100" namespace="monitoring" severity="warning"	FIRING	14m 4.309s	100

description	CPU load on host is over 50% Value = 100 Instance = 192.168.49.102:9100
summary	Host CPU load high

3. **Check Alerts in Alertmanager UI:**

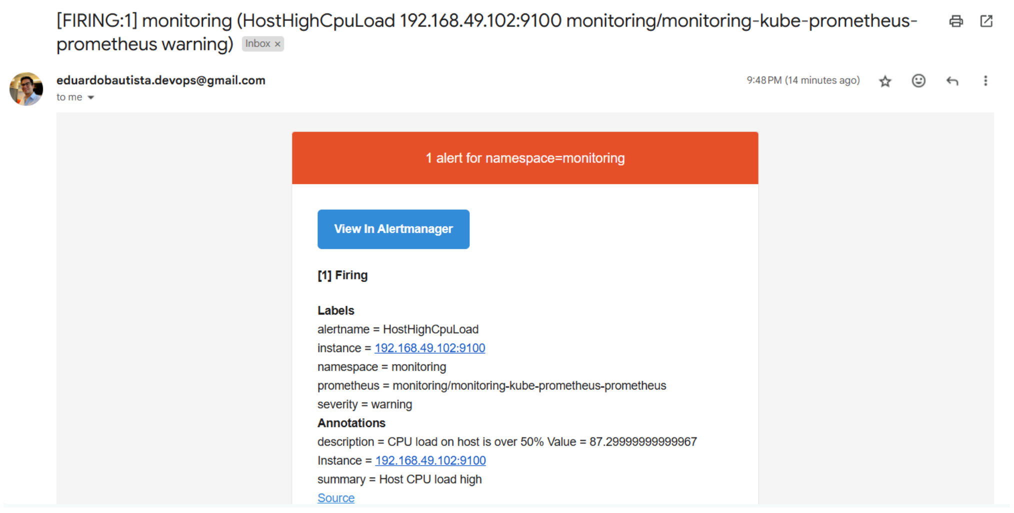

127.0.0.1:9093/api/v2/alerts

Pretty-print

HostHighCpuLoad 1/1

```
[{"annotations":{"description":"KubeScheduler has disappeared from Prometheus target discovery.","runbook_url":"https://runbooks.prometheus-operator.dev/runbooks/kubernetes/kubeschedule
discovery."},"endsAt":"2025-03-15T01:05:12.153Z","fingerprint":"1875b8db6d322e29","receivers":[{"name":"null"}],"startsAt":"2025-03-14T18:32:42.153Z","status":{"inhibitedBy":[],"mutedBy":[],"silencedBy":[],"state":"active"},"updatedAt":
15T01:01:12.155Z","generatorURL":"http://monitoring-kube-prometheus-prometheus.monitoring:9090/graph?g0.expr=absent%28up%7Bjob%3D%22kube-scheduler%22%7D+%3D%3D+1%29\u0026g0.tab=1","labels":{"alertname":"KubeSchedulerDown","prometheus":"
prometheus-prometheus","severity":"critical"}},{"annotations":{"description":"KubeControllerManager has disappeared from Prometheus target discovery.","runbook_url":"https://runbooks.prometheus-
operator.dev/runbooks/kubernetes/kubecontrollermanagerdown","summary":"Target disappeared from Prometheus target discovery."},"endsAt":"2025-03-15T01:05:08.745Z","fingerprint":"99beeeb88ac09d37","receivers":[{"name":"null"}],"startsAt":
14T18:32:38.745Z","status":{"inhibitedBy":[],"mutedBy":[],"silencedBy":[],"state":"active"},"updatedAt":"2025-03-15T01:01:08.748Z","generatorURL":"http://monitoring-kube-prometheus-prometheus.monitoring:9090/graph?g0.expr=absent%28up%7B
manager%22%7D+%3D%3D+1%29\u0026g0.tab=1","labels":{"alertname":"KubeControllerManagerDown","prometheus":"monitoring/monitoring-kube-prometheus-prometheus","severity":"critical"}},{"annotations":{"description":"This is an alert meant to
pipeline is functional.\nThis alert is always firing, therefore it should always be firing in Alertmanager\nand always fire against a receiver. There are integrations with various notification\nmechanisms that send a notification when t
example the\n\"DeadMansSnitch\" integration in PagerDuty.\n","runbook_url":"https://runbooks.prometheus-operator.dev/runbooks/general/watchdog","summary":"An alert that should always be firing to certify that Alertmanager is working pro
15T01:04:45.730Z","fingerprint":"d200352fd868ba95","receivers":[{"name":"null"}],"startsAt":"2025-03-14T18:17:15.730Z","status":{"inhibitedBy":[],"mutedBy":[],"silencedBy":[],"state":"active"},"updatedAt":"2025-03-15T01:00:45.734Z","gen
kube-prometheus-prometheus.monitoring:9090/graph?g0.expr=vector%281%29\u0026g0.tab=1","labels":{"alertname":"Watchdog","prometheus":"monitoring/monitoring-kube-prometheus-prometheus","severity":"none"}},{"annotations":{"description":"CP
Value = 99.08888888888922\n Instance = 192.168.49.102:9100\n","summary":"Host CPU load high"},"endsAt":"2025-03-15T01:05:27.764Z","fingerprint":"e8b047e66407d2c1","receivers":[{"name":"monitoring/main-rules-alert-config/email"}],"starts
15T00:47:57.764Z","status":{"inhibitedBy":[],"mutedBy":[],"silencedBy":[],"state":"active"},"updatedAt":"2025-03-15T01:01:27.766Z","generatorURL":"http://monitoring-kube-prometheus-prometheus.monitoring:9090/graph?g0.expr=100+-
%28avg+by+%28instance%29+%28rate%28node_cpu_seconds_total%7Bmode%3D%22idle%22%7D%5B2m%5D%29%29+%2A+100%29+%3E+50\u0026g0.tab=1","labels":{"alertname":"HostHighCpuLoad","instance":"192.168.49.102:9100","namespace":"monitoring","promethe
prometheus-prometheus","severity":"warning"}}]
```

4. **Email received:**


[FIRING:1] monitoring (HostHighCpuLoad 192.168.49.102:9100 monitoring/monitoring-kube-prometheus-prometheus warning)

eduardobautista.devops@gmail.com
to me
9:48 PM (14 minutes ago)

1 alert for namespace=monitoring

View In Alertmanager

[1] Firing

Labels
alertname = HostHighCpuLoad
instance = 192.168.49.102:9100
namespace = monitoring
prometheus = monitoring/monitoring-kube-prometheus-prometheus
severity = warning
Annotations
description = CPU load on host is over 50% Value = 87.29999999999967
Instance = 192.168.49.102:9100
summary = Host CPU load high
Source

Troubleshooting Email Issues

If emails are not received, check Alertmanager logs:

`kubectl logs alertmanager-monitoring-kube-prometheus-alertmanager-0 -n monitoring -c alertmanager`

Possible authentication errors should be investigated.
