## **1、后台启动**

cat > /usr/lib/systemd/system/prometheus.service << EOF

[Unit]

Description=prometheus server daemon

[Service]

Restart=on-failure

ExecStart=/usr/local/prometheus/prometheus --config.file=/usr/local/prometheus/prometheus.yml

[Install]

WantedBy=multi-user.target

EOF

systemctl daemon-reload

systemctl start prometheus.service

## **2、热更新**

prometherus 修改了配置文件支持热更新

kill -hup `ps -ef|grep prometheus|grep -v grep| awk '{print $2}'`

## **3、语法检查**

./promtool check config prometheus.yml

## **4、node-export**

cat > /usr/lib/systemd/system/node\_exporter.service << EOF

[Unit]

Description=node\_exporter daemon

[Service]

Restart=on-failure

ExecStart=/usr/local/node\_exporter/node\_exporter

[Install]

WantedBy=multi-user.target

EOF

systemctl daemon-reload

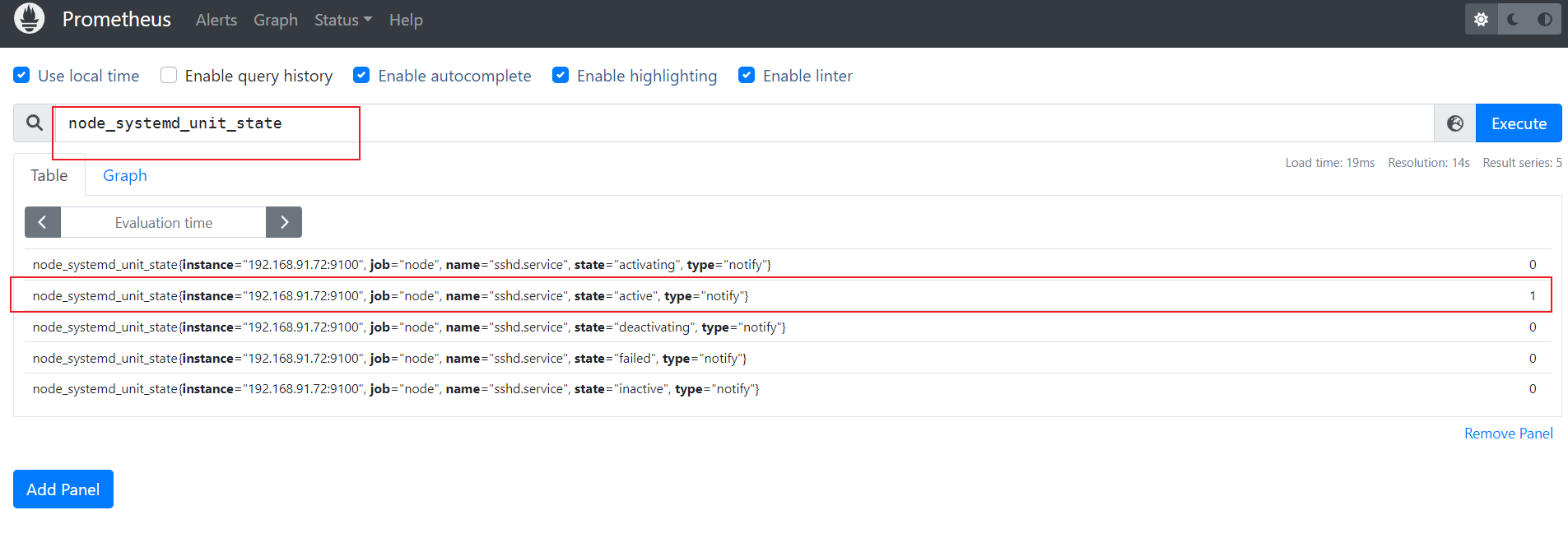
systemctl start node\_exporter.service

【监控服务运行状态】

添加 --collector.systemd --collector.systemd.unit-whitelist=(docker|sshd|nginx).service

ExecStart=/usr/local/node\_exporter/node\_exporter --collector.systemd --collector.systemd.unit-whitelist=(docker|sshd|nginx).service

查询 node\_systemd\_unit\_state



## **5、Grafana**

1. *主机基础监控(cpu，内存，磁盘，网络)*

9276

<https://grafana.com/grafana/dashboards/9276>

<https://grafana.com/grafana/dashboards/13332>

## **6、监控docker**

<https://github.com/google/cadvisor>

<https://grafana.com/grafana/dashboards/193>

**6.1 部署cadvisor**

|  |
| --- |
| sudo docker run \  --volume=/:/rootfs:ro \  --volume=/var/run:/var/run:ro \  --volume=/sys:/sys:ro \  --volume=/var/lib/docker/:/var/lib/docker:ro \  --volume=/dev/disk/:/dev/disk:ro \  --publish=8080:8080 \  --detach=true \  --name=cadvisor \  --privileged \  --device=/dev/kmsg \  google/cadvisor |

<http://192.168.91.72:8080/metrics>

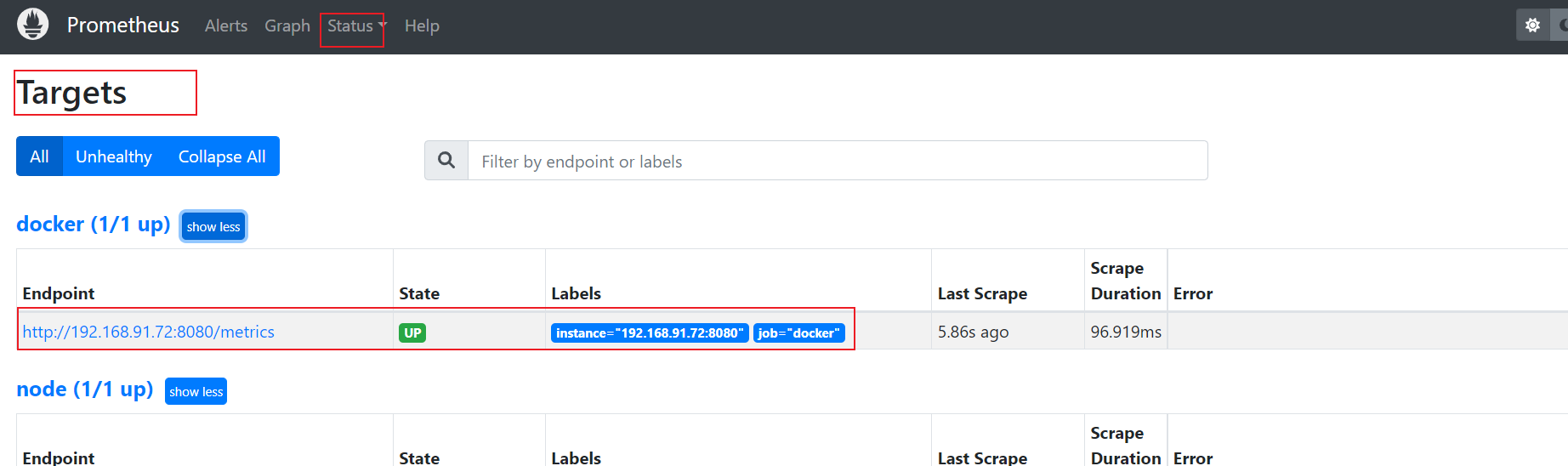


**6.2 编辑prometheus.yml**

|  |
| --- |
| - job\_name: "docker"  static\_configs:  - targets: ['192.168.91.72:8080'] #docker所在服务IP |

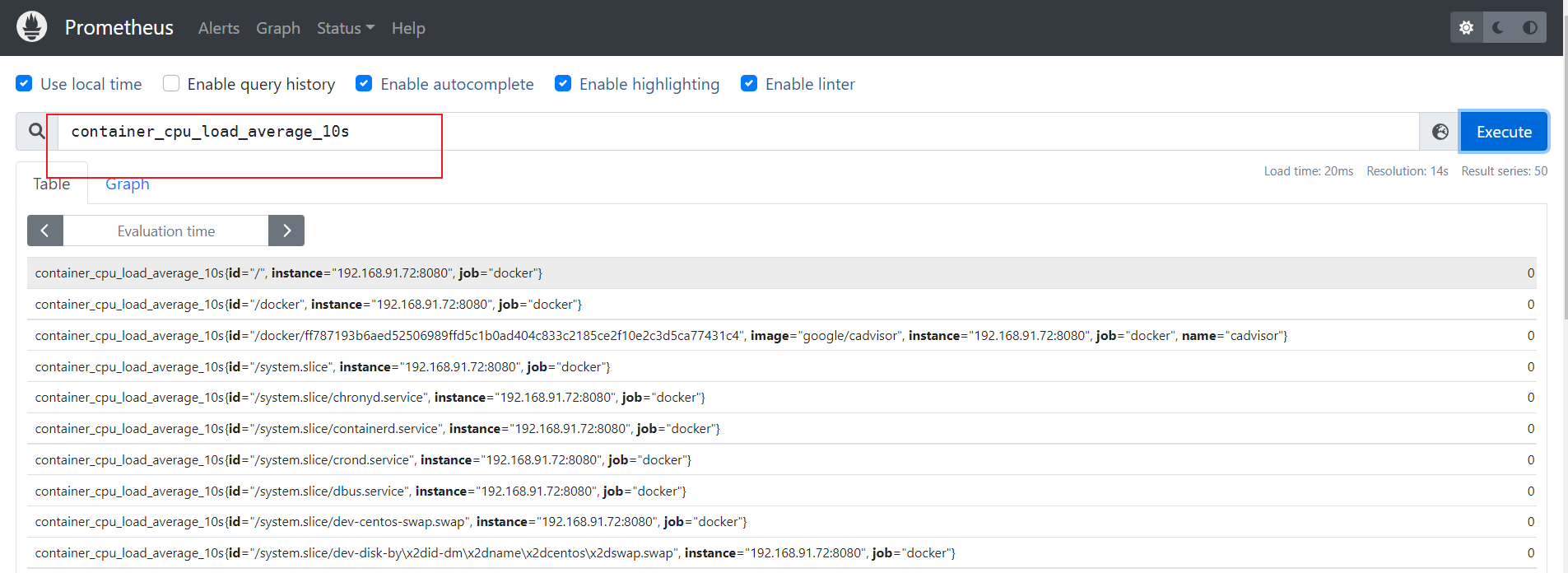
./promtool check config prometheus.yml

kill -hup `ps -ef|grep prometheus|grep -v grep| awk '{print $2}'`



**6.3 验证**

出现container开头的监控项



图表形式 Grafana导入193



## **7、监控MySQL服务器**

https://github.com/prometheus/mysqld\_exporter

https://grafana.com/dashboards/7362

**7.1 数据库创建用户并授权**

|  |
| --- |
| CREATE USER 'exporter'@'localhost' IDENTIFIED BY 'XXXXXXXX' WITH MAX\_USER\_CONNECTIONS 3;  GRANT PROCESS, REPLICATION CLIENT, SELECT ON \*.\* TO 'exporter'@'localhost'; |

MariaDB [(none)]> create user 'exporter'@'localhost' identified by '123.com';

MariaDB [(none)]> flush privileges;

**7.2 创建本地免密登陆**

cat > /usr/local/mysqld/.my.cnf <<

[client]

user=exporter

password=123.com

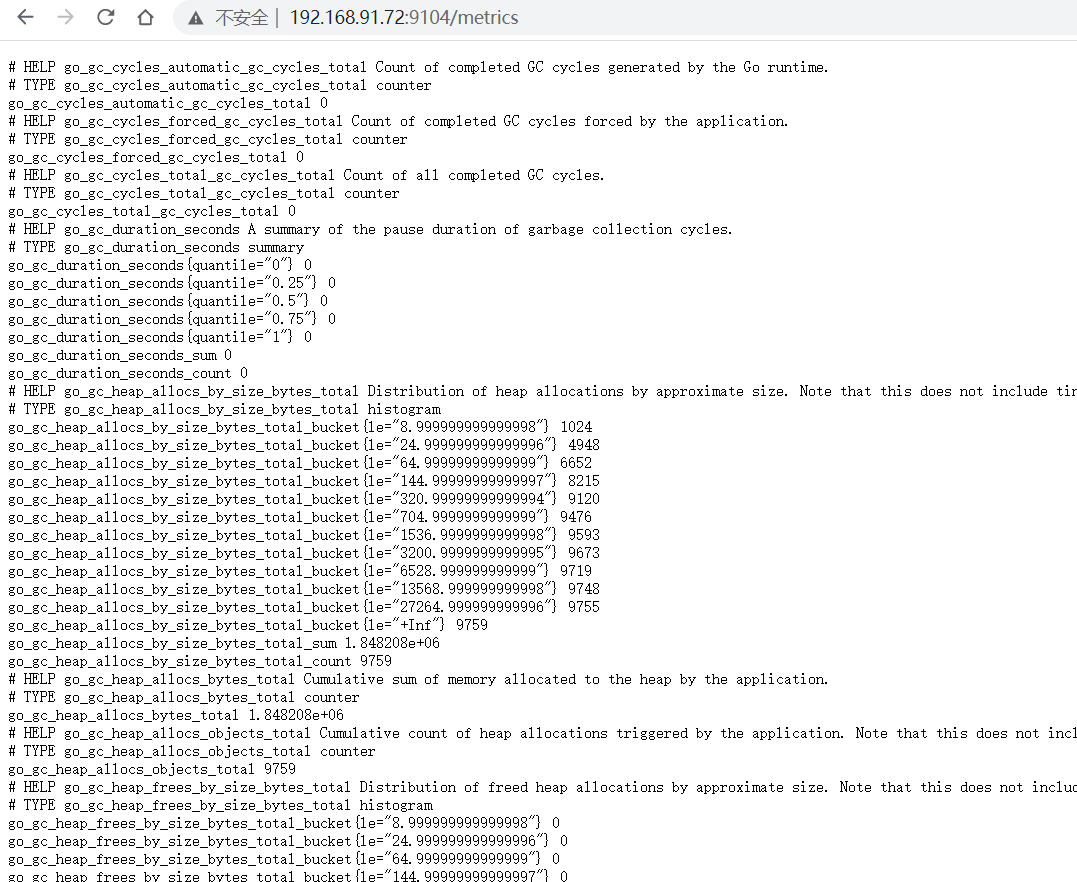
EOF

**7.3 启动mysqld\_exporter**

./mysqld\_exporter --config.my-cnf=.my.cnf

|  |
| --- |
| cat > /usr/lib/systemd/system/mysqld\_exporter << EOF  [Unit]  Description=mysqld\_exporter daemon  [Service]  Restart=on-failure  ExecStart=/usr/local/mysqld\_exporter/mysqld\_exporter --config.my-cnf=/usr/local/mysqld\_exporter/.my.cnf  [Install]  WantedBy=multi-user.target  EOF  systemctl daemon-reload  systemctl start mysqld\_exporter.service |

http://192.168.91.72:9104/metrics



**7.4 server 的 prometheus.yml 添加以下内容**

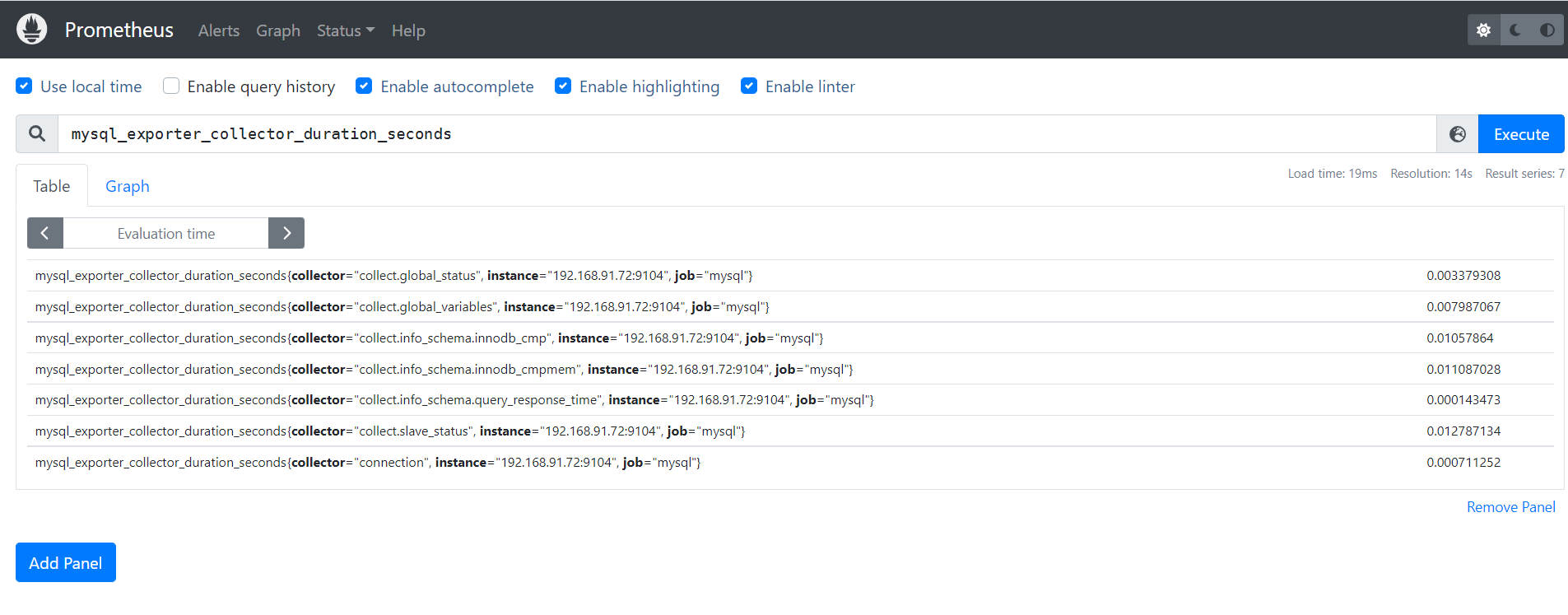
|  |
| --- |
| - job\_name: "mysql"  static\_configs:  - targets: ['192.168.91.72:9104'] |

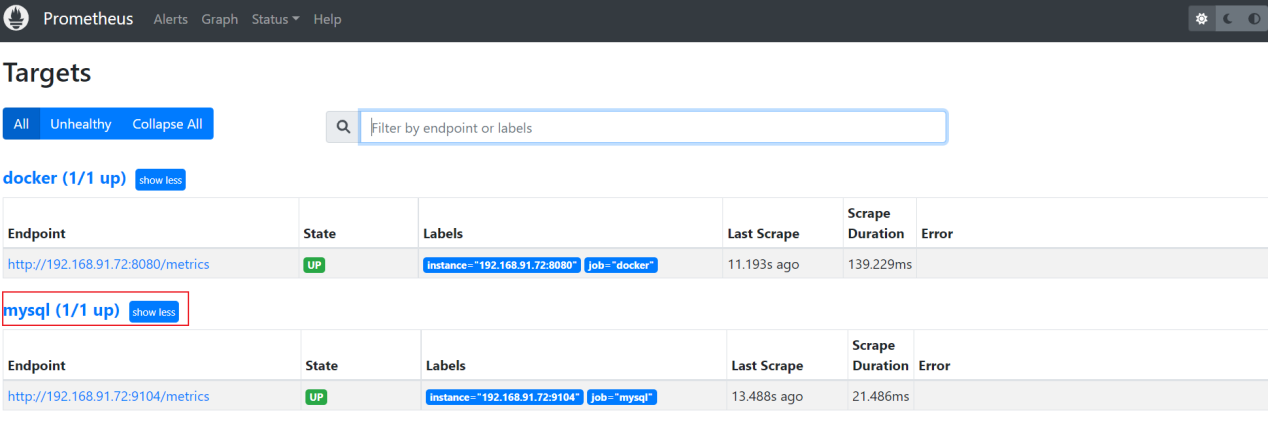
./promtool check config prometheus.yml

kill -hup `ps -ef|grep prometheus|grep -v grep| awk '{print $2}'`

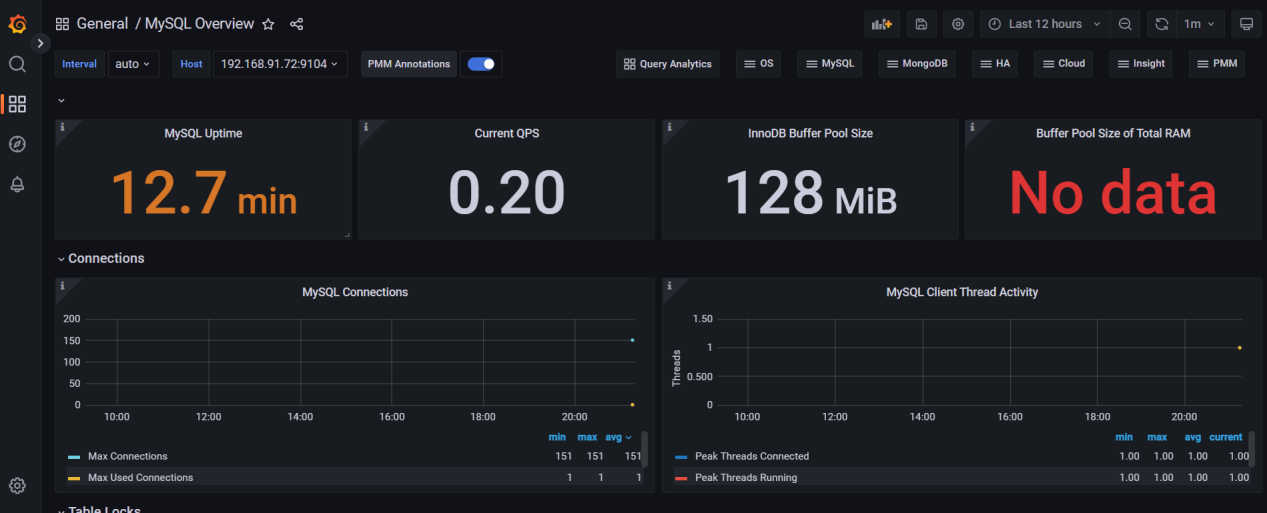
**7.5 验证**

Mysql开头的监控项





图表形式 Grafana导入7362



# **8、部署Alertmanager**

地址1：https://prometheus.io/download/

地址2：https://github.com/prometheus/alertmanager/releases

在prometheus server安装，（并不一定要安装在prometheus server）

1. 部署Alertmanager

2. 配置Prometheus与Alertmanager通信

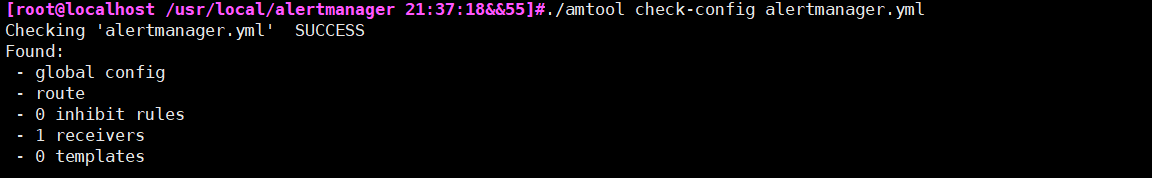
3. 在Prometheus中创建告警规则

**编辑alertmanager.yml**

|  |
| --- |
| global:    resolve\_timeout: 5m    smtp\_smarthost: 'smtp.qq.com:465'    smtp\_from: '751536336@qq.com'    smtp\_auth\_username: '751536336@qq.com'    smtp\_auth\_password: 'jjueigohmavibcbi'    smtp\_require\_tls: false  route:    group\_by: ['alertname']    group\_wait: 10s    group\_interval: 10s    repeat\_interval: 1m    receiver: 'mail'  receivers:    - name: 'mail'      email\_configs:        - to: '751536336@qq.com'  *#inhibit\_rules:*  *#  - source\_match:*  *#      severity: 'critical'*  *#    target\_match:*  *#      severity: 'warning'*  *#    equal: ['alertname', 'dev', 'instance']* |

**语法检查**

#./amtool check-config alertmanager.yml

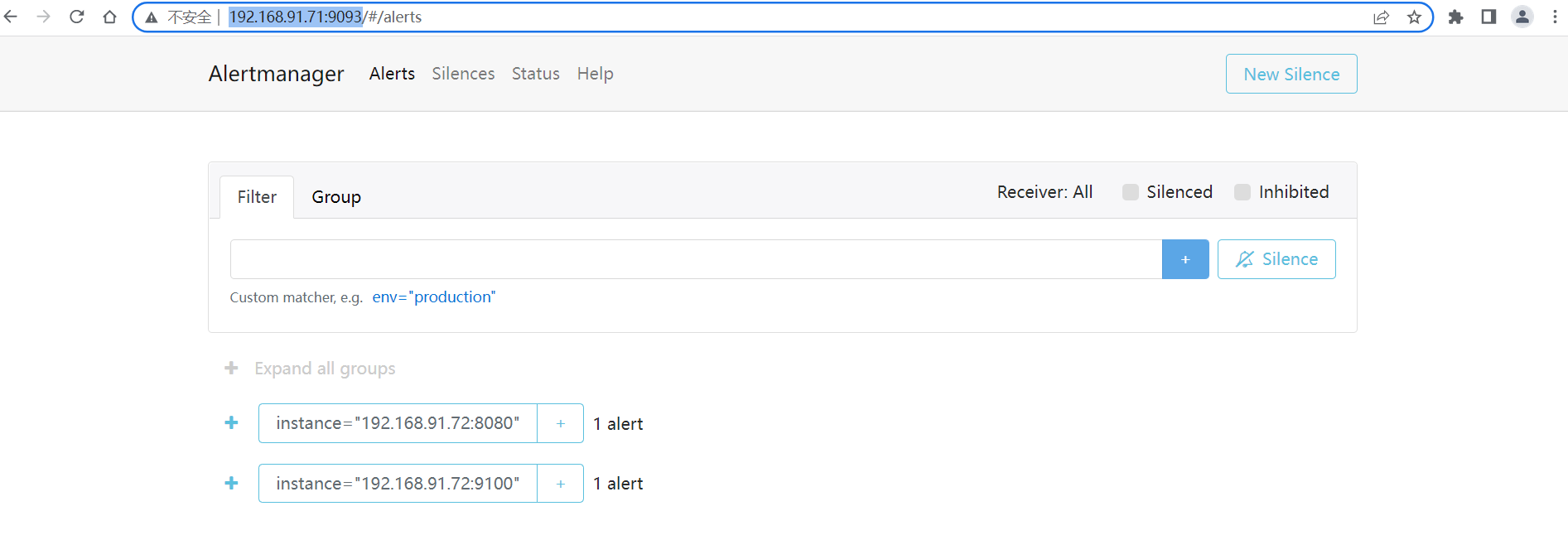


**启动alertmanager**

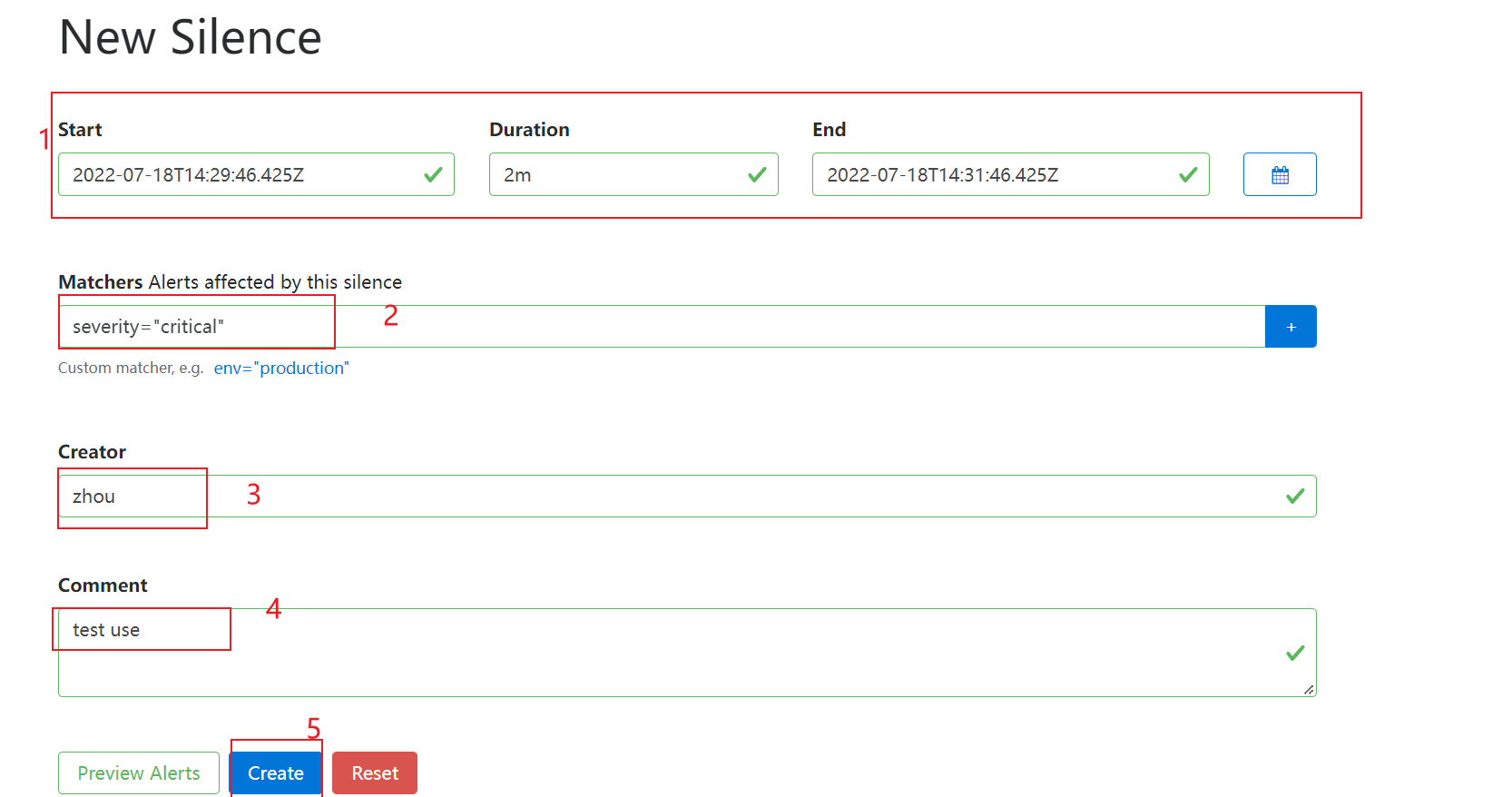
./alertmanager --config.file=alertmanager.yml

|  |
| --- |
| cat > /usr/lib/systemd/system/alertmanager.service << EOF  [Unit]  Description=alertmanager server daemon  [Service]  Restart=on-failure  ExecStart=/usr/local/alertmanager/alertmanager --config.file=/usr/local/alertmanager/alertmanager.yml  [Install]  WantedBy=multi-user.target  EOF  systemctl daemon-reload  systemctl start alertmanager.service |

<http://192.168.91.71:9093/>



**静默设置**



**编辑prometheus.yml**

**配置**alertmanager通信地址

|  |
| --- |
| alerting:  alertmanagers:  - static\_configs:  - targets:  - 127.0.0.1:9093 #修改此项。由于部署在本机 |

配置告警规则

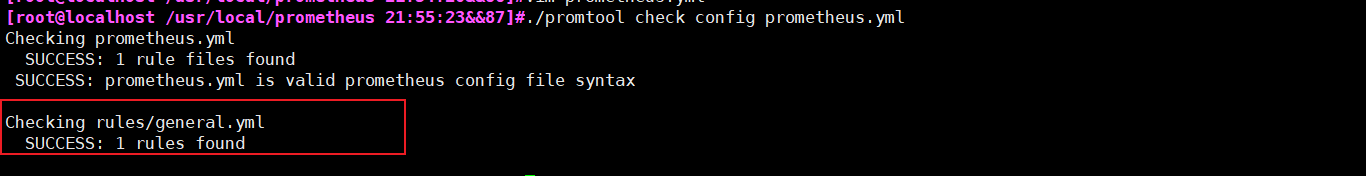
|  |
| --- |
| rule\_files:  - "rules/\*.yml" #当前目录下的rules，注意是yml部署yaml  #- "second\_rules.yml" |

mkdir rules

vim rules/general.yml

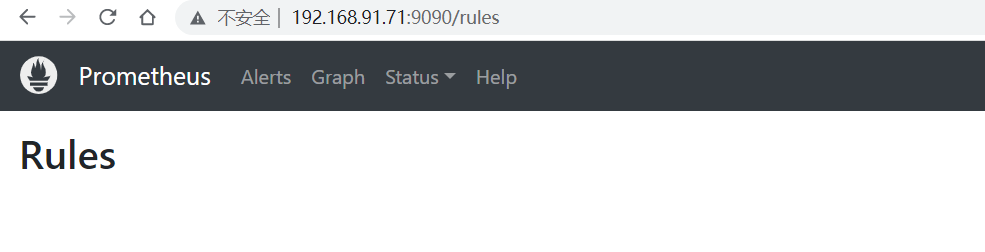
|  |
| --- |
| groups:  - name: general.rules    rules:    - alert: InstanceDown      expr: up == 0      for: 2m      labels:        severity: error      annotations:        summary: "Instance {{ $labels.instance }} 停止工作"        description: "{{ $labels.instance }}: job {{ $labels.job }} 已经停止5分钟以上." |

./promtool check config prometheus.yml



kill -hup `ps -ef|grep prometheus|grep -v grep| awk '{print $2}'`

没有添加之前rules是空的



添加之后rules

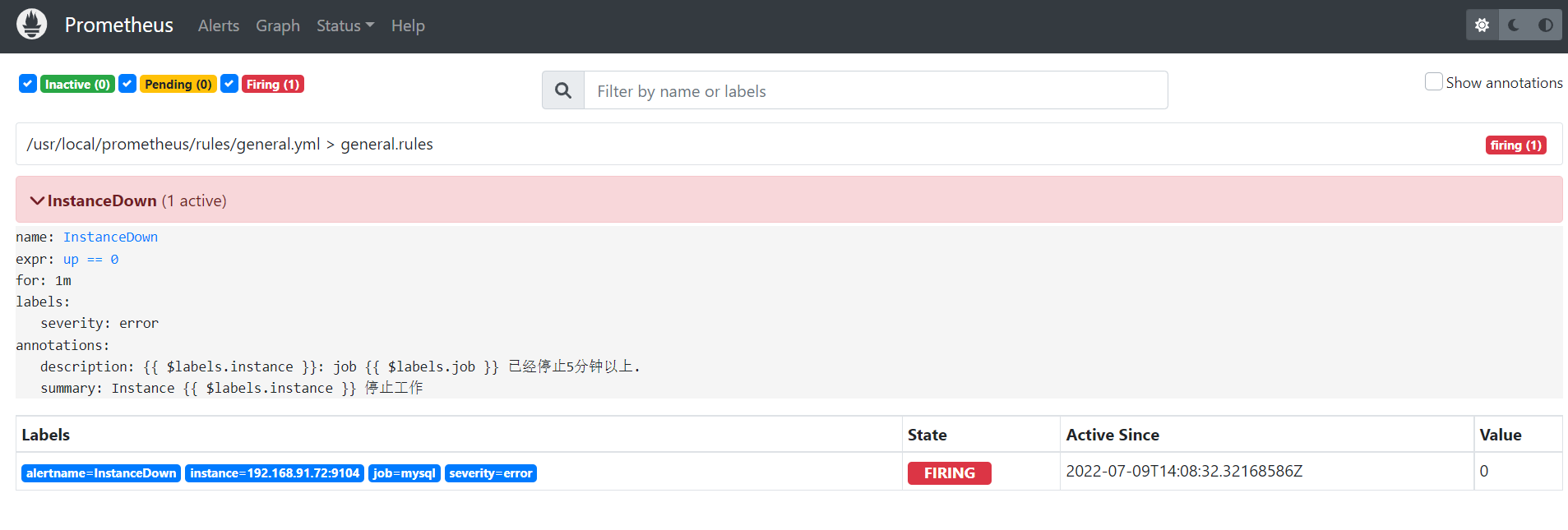


验证

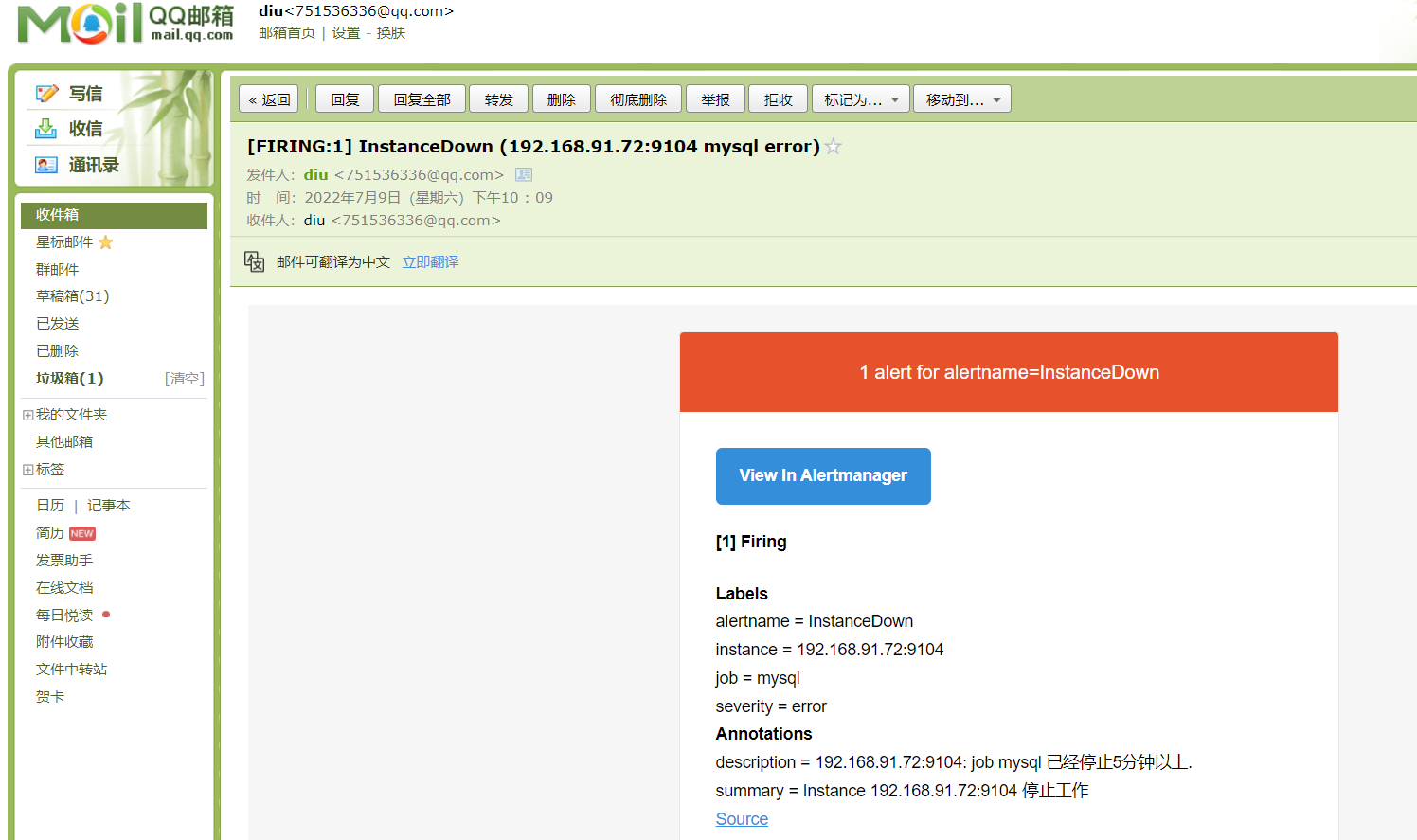
将mariadb停掉

./mysqld\_exporter --config.my-cnf=.my.cnf

Ctrl+c



邮箱收到报警



**告警状态**

Inactive: 这里什么都没有发生

Pending： 已出发阈值。但未满足告警持续时间

Firing: 已触发阈值且满足告警持续时间。警报发送给接受者。

vim /usr/local/prometheus/node.yml

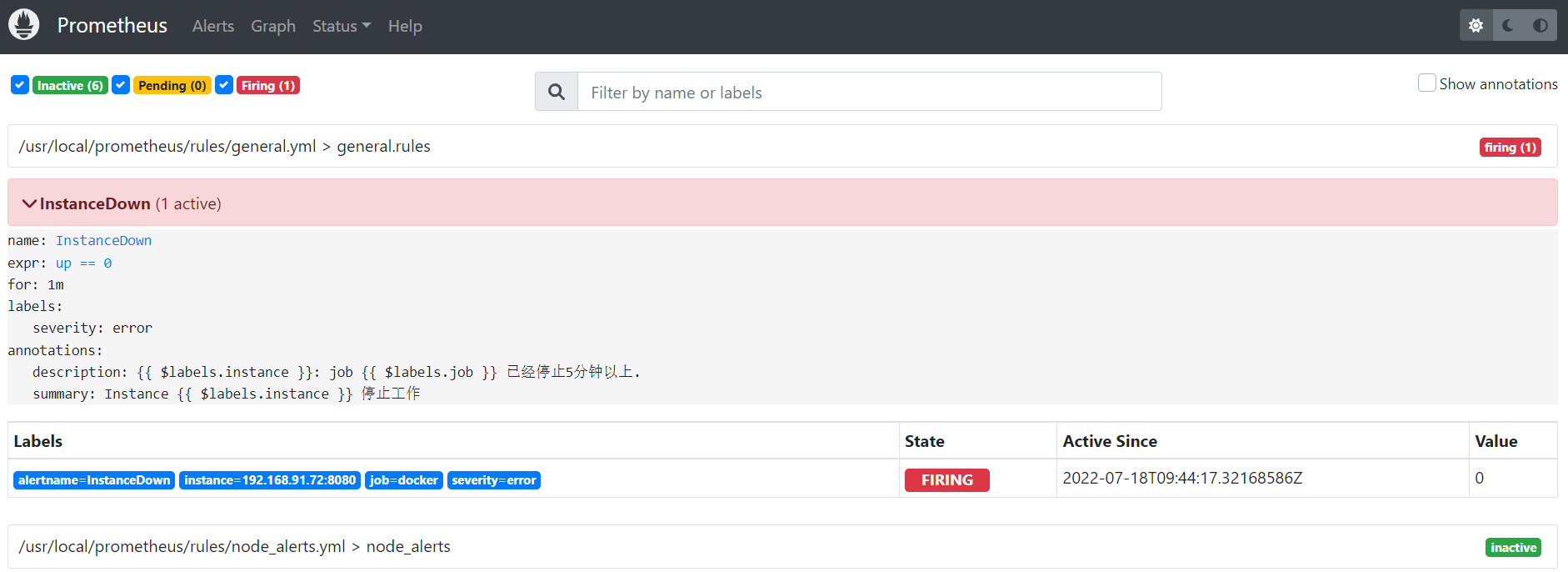
|  |
| --- |
| groups:  - name: node.rules    rules:    - alert: NodeFilesystemUsage      expr: 100 - (node\_filesystem\_free\_bytes{fstype=~"ext4|xfs"} / node\_filesystem\_size\_bytes{fstype=~"ext4|xfs"} \* 100) > 80      for: 2m      labels:        severity: warning      annotations:        summary: "{{$labels.instance}}: {{$labels.mountpoint }} 分区使用过高"        description: "{{$labels.instance}}: {{$labels.mountpoint }} 分区使用大于 80% (当前值: {{ $value }})"    - alert: NodeMemoryUsage      expr: 100 - (node\_memory\_MemFree\_bytes+node\_memory\_Cached\_bytes+node\_memory\_Buffers\_bytes) / node\_memory\_MemTotal\_bytes \* 100 > 80      for: 2m      labels:        severity: warning      annotations:        summary: "{{$labels.instance}}: 内存使用过高"        description: "{{$labels.instance}}: 内存使用大于 80% (当前值: {{ $value }})"    - alert: NodeCPUUsage      expr: 100 - (avg(irate(node\_cpu\_seconds\_total{mode="idle"}[5m])) by (instance) \* 100) > 80      for: 2m      labels:        severity: warning      annotations:        summary: "{{$labels.instance}}: CPU使用过高"        description: "{{$labels.instance}}: CPU使用大于 80% (当前值: {{ $value }})" |

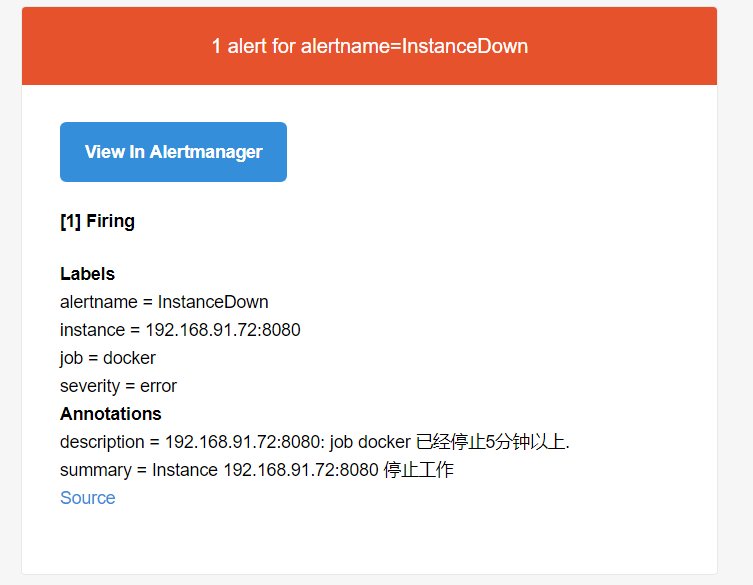
**实例1**

vim rules/node\_alerts.yml

|  |
| --- |
| groups:  - name: node\_alerts  rules:  - alert: HighNodeCPU  expr: instance:node\_cpu:avg\_rate5m > 4  for: 2m  labels:  severity: warning  annotations:  summary: High Node CPU for 1 hour  console: Thank you Test    - alert: DiskWillFillIn4Hours  expr: predict\_linear(node\_filesystem\_free\_bytes{mountpoint="/"}[1h], 4\*3600) < 0  for: 5m  labels:  severity: critical  annotations:  summary: Disk on {{ $labels.instance }} will fill in approximately 4 hours.  - alert: InstanceDown  expr: up{job="node"} == 0  for: 1m  labels:  severity: critical  annotations:  summary: Host {{ $labels.instance }} of {{ $labels.job }} is Down! |

docker stop cadvisor



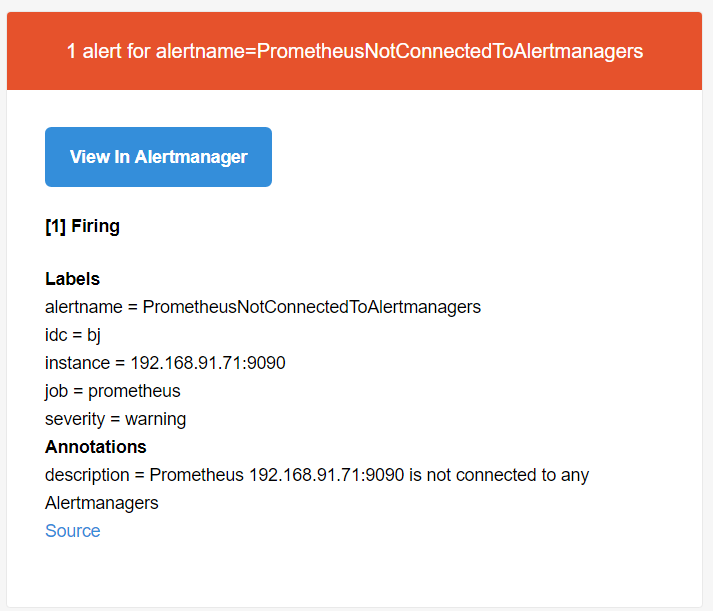


**实例2 添加 Prometheus 告警**

vim rules/prometheus\_alertes.yml

|  |
| --- |
| groups:  - name: prometheus\_alerts  rules:  - alert: PrometheusConfigReloadFailed  expr: prometheus\_config\_last\_reload\_successful == 0  for: 1m  labels:  severity: warning  annotations:  description: Reloading Prometheus configuration has failed on {{ $labels.instance }}.  - alert: PrometheusNotConnectedToAlertmanagers  expr: prometheus\_notifications\_alertmanagers\_discovered < 2  for: 1m  labels:  severity: warning  annotations:  description: Prometheus {{ $labels.instance }} is not connected to any Alertmanagers |

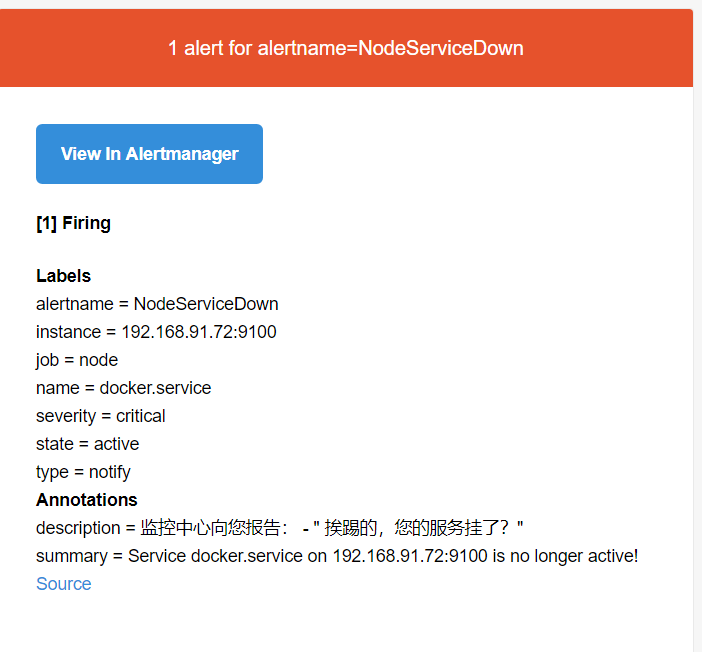
./promtool check config ./prometheus.yml



**实例3 添加 systemd 服务告警**

vi rules/service\_alertes.yml

|  |
| --- |
| groups:  - name: service\_alerts  rules:  - alert: NodeServiceDown  expr: node\_systemd\_unit\_state{state="active"} != 1  for: 10s  labels:  severity: critical  annotations:  summary: Service {{ $labels.name }} on {{ $labels.instance }} is no longer active!  description: 监控中心向您报告： - " 挨踢的，您的服务挂了？" |



**实例4 Altermanager 路由**

vim /usr/local/alertmanager/alertmanager.yml

|  |
| --- |
| global:  resolve\_timeout: 5m  smtp\_smarthost: 'smtp.qq.com:465'  smtp\_from: '751536336@qq.com'  smtp\_auth\_username: '751536336@qq.com'  smtp\_auth\_password: 'jjueigohmavibcbi'  smtp\_require\_tls: false  route:  group\_by: ['instance']  group\_wait: 30s  group\_interval: 5m  repeat\_interval: 3h  receiver: email  routes:  - match:  severity: critical  receiver: pager  - match\_re:  severity: ^(warning|critical)$  receiver: support\_team    receivers:  - name: 'email'  email\_configs:  - to: '751536336@qq.com'  - name: 'support\_team'  email\_configs:  - to: '15280808129@163.com'  - name: 'pager'  email\_configs:  - to: '15280808129@139.com' |

蓝色标记为添加内容

./amtool check-config alertmanager.yml

# **9、全方位监控k8s**

k apply -f nfs.yml

k apply -f prometheus-rbac.yaml

k apply -f prometheus-configmap.yaml

k apply -f prometheus-rules.yaml

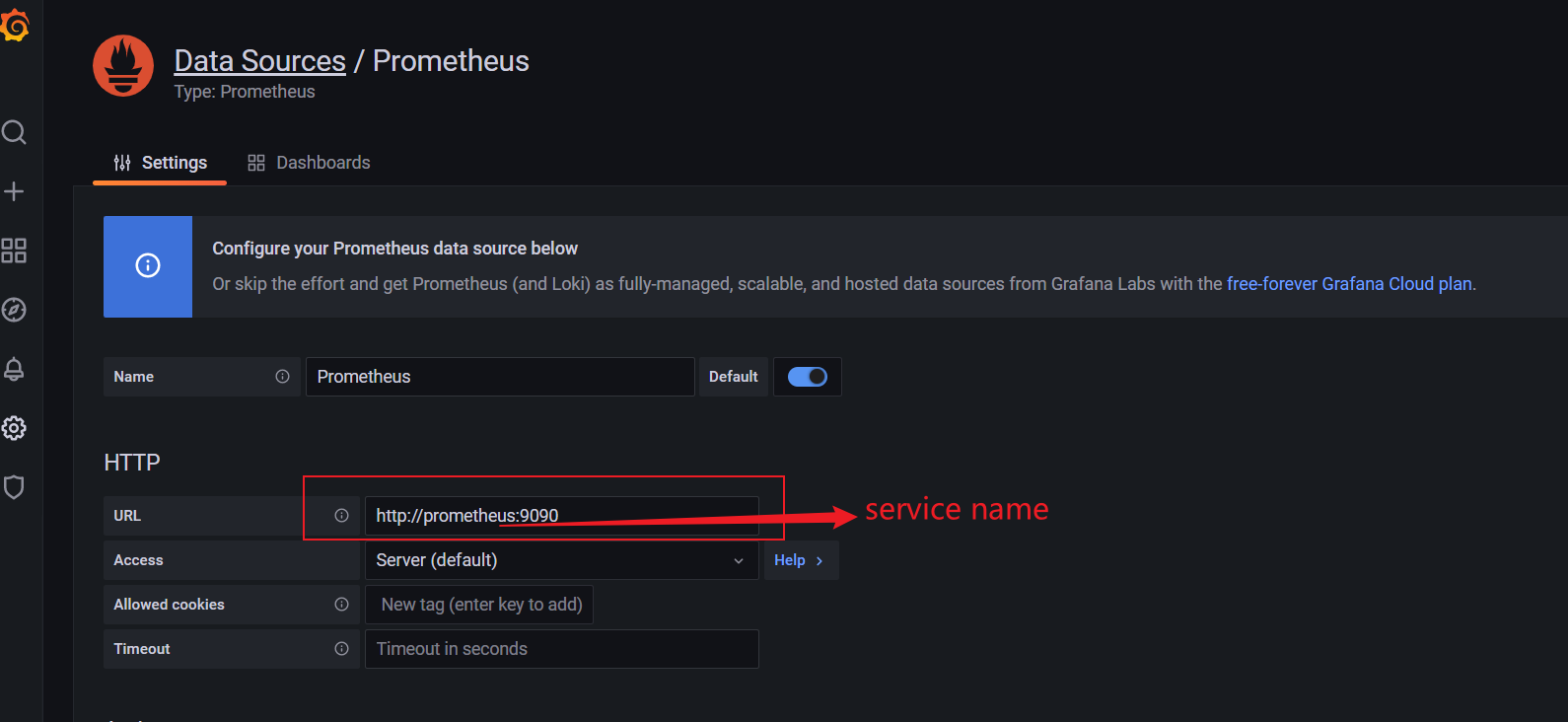
k apply -f prometheus-statefulset.yaml

k apply -f prometheus-service.yaml

# **10、在K8S中部署Grafana与可视化**

k apply -f grafana.yaml

添加数据源



Grafana是一个开源的度量分析和可视化系统。

https://grafana.com/grafana/download

推荐模板：

• 集群资源监控：3119

• 资源状态监控 ：6417

• Node监控 ：9276

# **11、监控K8S集群Node**

node\_exporter：用于\*NIX系统监控，使用Go语言编写的收集器。

使用文档：https://prometheus.io/docs/guides/node-exporter/

GitHub：https://github.com/prometheus/node\_exporter

exporter列表：https://prometheus.io/docs/instrumenting/exporters/

sh node\_exporter.sh

|  |
| --- |
| #!/bin/bash  wget https://github.com/prometheus/node\_exporter/releases/download/v1.3.1/node\_exporter-1.3.1.linux-amd64.tar.gz  tar zxf node\_exporter-1.3.1.linux-amd64.tar.gz  mv node\_exporter-1.3.1.linux-amd64 /usr/local/node\_exporter  cat <<EOF >/usr/lib/systemd/system/node\_exporter.service  [Unit]  Description=https://prometheus.io  [Service]  Restart=on-failure  ExecStart=/usr/local/node\_exporter/node\_exporter --collector.systemd --collector.systemd.unit-whitelist=(docker|kubelet|kube-proxy|flanneld).service  [Install]  WantedBy=multi-user.target  EOF  systemctl daemon-reload  systemctl enable node\_exporter  systemctl restart node\_exporter |

编辑vim prometheus-configmap.yaml

|  |
| --- |
| - job\_name: kubernetes-nodes  scrape\_interval: 30s  static\_configs:  - targets:  - 192.168.91.100:9100  - 192.168.91.101:9100  - 192.168.91.102:9100 |

Grafana 导入 9276模板

# **12、监控K8S资源对象**

kube-state-metrics采集了k8s中各种资源对象的状态信息：

kube\_daemonset\_\*

kube\_deployment\_\*

kube\_job\_\*

kube\_namespace\_\*

kube\_node\_\*

kube\_persistentvolumeclaim\_\*

kube\_pod\_container\_\*

kube\_pod\_\*

kube\_replicaset\_\*

kube\_service\_\*

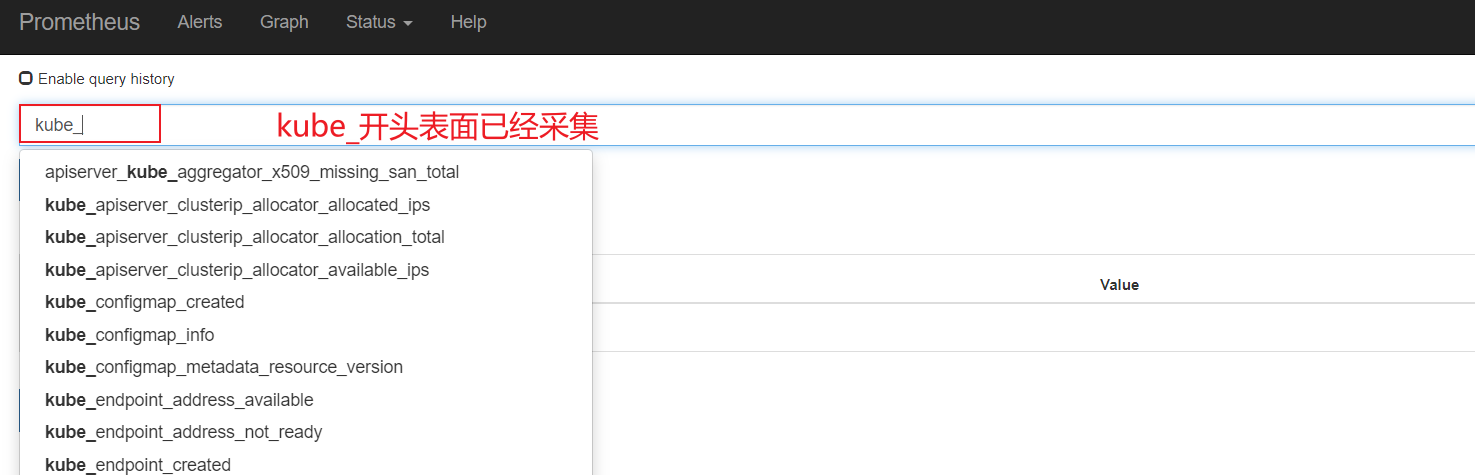
kube\_statefulset\_\*

<https://github.com/kubernetes/kube-state-metrics>

**部署**

|  |
| --- |
| k apply -f kube-state-metrics-rbac.yaml  k apply -f kube-state-metrics-deployment.yaml  k apply -f kube-state-metrics-service.yaml |

**验证**

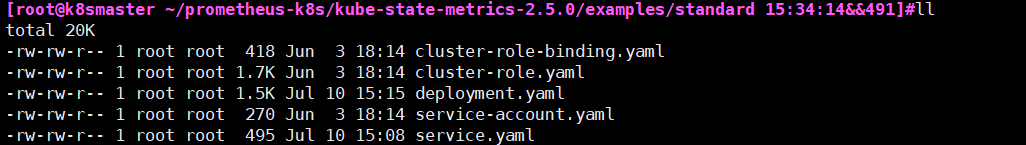


Grafana 导入模板6417

**二**

wget https://github.com/kubernetes/kube-state-metrics/archive/refs/tags/v2.5.0.tar.gz

cd kube-state-metrics-2.5.0/examples/standard



vim service.yaml

添加如下内容 *#添加此参数，允许prometheus自动发现*

|  |
| --- |
| annotations:  prometheus.io/scrape: 'true' |

### **创建**

kubectl apply -f ./

### **查看是否运行成功**

kubectl get pod -n kube-system -owide |grep kube-state-metrics



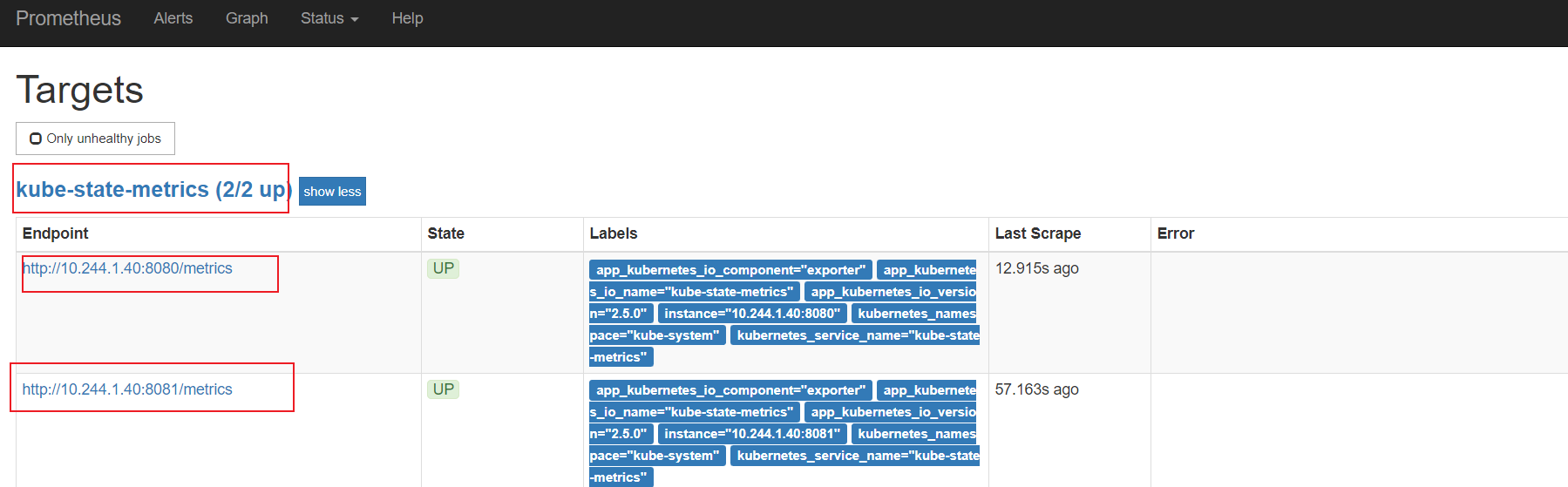
kubectl get svc -n kube-system |grep kube-state-metrics

# 验证指标是否采集成功 请求kube-state-metrics的pod ip+8080端口

curl 10.244.1.40:8080/metrics

编辑prometheus-configmap.yaml

|  |
| --- |
| - job\_name: 'kube-state-metrics'  scheme: http  bearer\_token\_file: /var/run/secrets/kubernetes.io/serviceaccount/token  metrics\_path: metrics  tls\_config:  insecure\_skip\_verify: true  kubernetes\_sd\_configs:  - api\_server: https://192.168.91.100:6443  role: pod  tls\_config:  insecure\_skip\_verify: true  bearer\_token\_file: /var/run/secrets/kubernetes.io/serviceaccount/token  relabel\_configs:  - source\_labels: [\_\_meta\_kubernetes\_pod\_name]  action: replace  target\_label: pod  - action: labelmap  regex: \_\_meta\_kubernetes\_pod\_label\_(.+)  - source\_labels: [\_\_meta\_kubernetes\_pod\_ip]  regex: (.+)  target\_label: \_\_address\_\_  replacement: ${1}:8080  - source\_labels: ["\_\_meta\_kubernetes\_pod\_container\_name"]  regex: "^kube-state-metrics.\*"  action: keep |



<https://www.cnblogs.com/suyj/p/16053993.html>

<https://blog.csdn.net/qq_36464836/article/details/112918635>

Grafana 导入13332 13105模板

# **在K8S中部署Alertmanager**

1. 部署Alertmanager

2. 配置Prometheus与Alertmanager通信

3. 配置告警

1. prometheus指定rules目录

2. configmap存储告警规则

3. configmap挂载到容器rules目录

4. 增加alertmanager告警配置

**部署**

|  |
| --- |
| k apply -f alertmanager-configmap.yaml  k apply -f alertmanager-pvc.yaml  k apply -f alertmanager-deployment.yaml  k apply -f alertmanager-service.yaml |

**验证**

systemctl stop node\_exporter.service 任意一台node

等待发送报警