**DevOps Case Study**

Below are some questions we would like you to work on. We expect to receive your feedback within 5 business days. Please feel free to contact us if you found the questions are unclear.

Configuration management

Suggested environment: Ubuntu 20 LTS, ansible 2.9.16, puppet 4.5 or above.

1. Which ansible command can display all ansible\_ configuration for a host.

ansible-config dump

1. Please configure a cron job that runs logrotate on all machines every 10 minutes between 2h - 4h.

# crontab -e -u ubuntu

\*/10 2-3 \* \* \* /usr/sbin/logrotate /etc/logrotate.conf

1. Please deploy ntpd package to the following 3 servers:

app-vm1.fra1.internal (192.168.0.2)

db-vm1.fra1.db (192.168.0.3)

web-vm1.fra1.web (192.168.0.4)

with custom config of /etc/ntpd.conf:



We also need to deploy monitoring template onto our nagios server “monitoring.fra1.internal”, each of the above machines should use the following nagios templates:



**Solution:**

# mkdir ntp-servers/config -p

# cd ntp-servers

# vim hosts

[ntp\_servers]

app-vm1.fra1.internal ansible\_host=192.168.0.2

db-vm1.fra1.db ansible\_host=192.168.0.3

web-vm1.fra1.web ansible\_host=192.168.0.4

[nagios\_server]

monitoring.fra1.internal ansible\_host=192.168.0.5

# vim config/ntpd.conf

tinker panic 0

restrict default kod nomodify notrap nopeer noquery

restrict-6 default kod nomodify notrap nopeer noquery

restrict 127.0.0.1

restrict-6::1

server 192.168.0.252 minpoll 4 maxpoll 8

server 192.168.0.251 minpoll 4 maxpoll 8

server 192.168.0.0 # local clock

fudge 192.168.0.0 stratum 10

driftfile /var/lib/ntp/drift

keys /etc/ntp/keys

# vim deploy\_ntpd.yml

---

- name: Deploy ntp server with custom configuration

hosts: ntp\_servers

become: yes

tasks:

- name: Install ntp package on Ubuntu

apt:

name: ntp

state: present

update\_cache: yes

when: ansible\_os\_family == "Debian"

- name: Deploy custom ntpd.conf

copy:

src: config/ntpd.conf

dest: /etc/ntpd.conf

owner: root

group: root

mode: '0644'

- name: Ensure ntp is running and enabled on Ubuntu

service:

name: ntp

state: started

enabled: yes

when: ansible\_os\_family == "Debian"

ansible-playbook -i hosts deploy\_ntpd.yml --syntax-check

ansible-playbook -i hosts deploy\_ntpd.yml

# vim deploy\_nagios.yml

# ---

# - name: Deploy Nagios monitoring templates

# hosts: nagios\_server

# become: yes

# tasks:

# - name: Create Nagios host configuration

# copy:

# content: |

# define host {

# host\_name {{ item.name }}

# address {{ item.ip }}

# check\_command check-ping

# active\_checks\_enabled 1

# passive\_checks\_enabled 1

# }

# dest: /etc/nagios/conf.d/{{ item.name }}.cfg

# loop:

# - { name: 'app-vm1.fra1.internal', ip: '192.168.0.2' }

# - { name: 'db-vm1.fra1.db', ip: '192.168.0.3' }

# - { name: 'web-vm1.fra1.web', ip: '192.168.0.4' }

# - name: Create Nagios service configuration

# copy:

# content: |

# define service {

# service\_description ntp\_process

# host\_name {{ item.name }}

# check\_command check\_ntp

# check\_interval 10

# }

# dest: /etc/nagios/conf.d/{{ item.name }}\_service.cfg

# loop:

# - { name: 'app-vm1.fra1.internal' }

# - { name: 'db-vm1.fra1.db' }

# - { name: 'web-vm1.fra1.web' }

# - name: Restart Nagios service

# service:

# name: nagios

# state: restarted

# ansible-playbook -i hosts deploy\_nagios.yml --syntax-check

# ansible-playbook -i hosts deploy\_nagios.yml

# Docker/Kubernetes

Suggested environment: Ubuntu 20 LTS, docker 19 or above

1. Prepare a docker-compose for a nginx server.

Requirements:

* nginx logs need to survive between nginx container restarts
* docker should use network bridge subnet 172.20.8.0/24

**Solution:**

mkdir nginx-server

cd nginx-server

vim docker-compose.yml

services:

nginx:

image: nginx:latest

ports:

- "80:80"

volumes:

- ./nginx.conf:/etc/nginx/nginx.conf

- ./logs:/var/log/nginx

networks:

custom\_network:

ipv4\_address: 172.20.8.2

networks:

custom\_network:

driver: bridge

ipam:

config:

- subnet: 172.20.8.0/24

vim nginx.conf

events {

worker\_connections 100;

}

http {

server {

listen 80;

server\_name localhost;

location / {

root /usr/share/nginx/html;

index index.html index.htm;

}

error\_log /var/log/nginx/error.log;

access\_log /var/log/nginx/access.log;

}

}

docker compose config

docker compose up -d

docker compose ps -a

docker container inspect nginx-server-nginx-1 | grep "IPAddress"

1. Which Kubernetes command you will use to identify the reason for a pod restart in the project "internal" under namespace "production".

**Solution:**

kubectl describe pods internal -n production

1. Consider the followings:

POD NAME CPU(cores) MEMORY(bytes)

java-app-7d9d44ccbf-lmvbc java-app 3m 951Mi

java-app-7d9d44ccbf-lmvbc java-app-logrotate 1m 45Mi

java-app-7d9d44ccbf-lmvbc java-app-fluentd 1m 84Mi

java-app-7d9d44ccbf-lmvbc mongos 4m 62Mi

Application pod has the following resource quota:

* Memory request & limit: 1000 & 1500
* CPU request & limit: 1000 & 2000
* Xmx of 1000M

Java-app keep restarting at random. From Kubernetes configuration perspective, what are the possible reasons for the pod restarts?

**Possible Reasons:**

**Resource Limits:** If a pod consumes too many resources, it might be terminated and restarted by Kubernetes. This can happen due to an Out of Memory (OOM) error.

**Failed Liveness Probes:** If the liveness probe fails, Kubernetes will restart the pod to try to recover it.

**Application Crashes:** If the application inside the pod crashes frequently, Kubernetes will keep restarting it, leading to a CrashLoopBackOff state.

**Failed Image Pulls:** If Kubernetes cannot pull the container image, it might cause the pod to restart.

# Helm

Please use the accompanied elasticsearch helm template to create a Kubernetes deployment of elasticsearch.

Provide a screenshot & deployment yaml of the resultant deployment in Kubernetes.

**Solution:**

asraful@cks-master:~/DevOps-assessment/DevOps/elasticsearch\_helm/elasticsearch\_helm/elasticsearch$ helm lint .

==> Linting .

[INFO] Chart.yaml: icon is recommended

1 chart(s) linted, 0 chart(s) failed

asraful@cks-master:~/DevOps-assessment/DevOps/elasticsearch\_helm/elasticsearch\_helm$ helm install elasticserach ./elasticsearch/ -f elasticsearch/envs/qa/customer-abc-elasticsearch/values.yaml

NAME: elasticserach

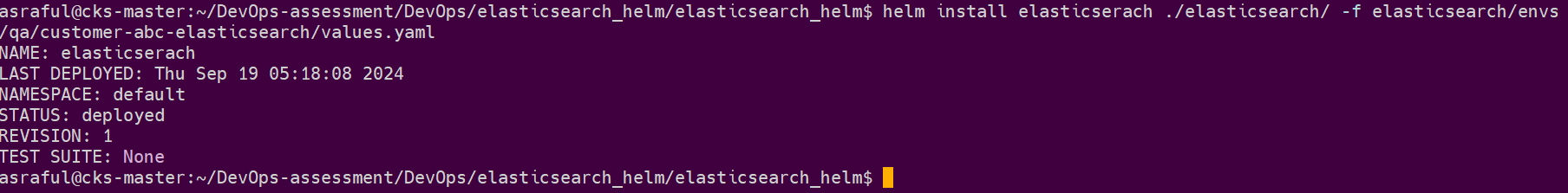
LAST DEPLOYED: Thu Sep 19 05:18:08 2024

NAMESPACE: default

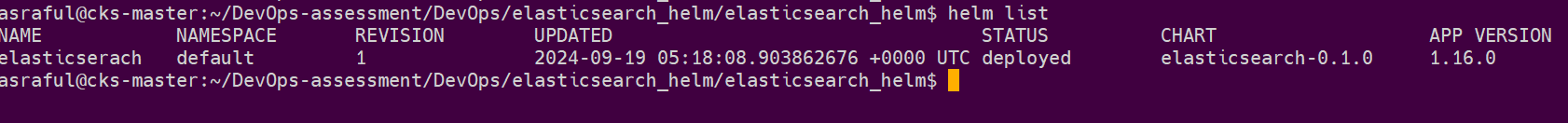
STATUS: deployed

REVISION: 1

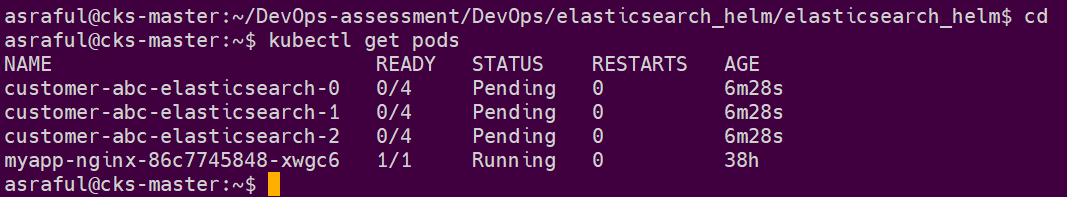
TEST SUITE: None



asraful@cks-master:~/DevOps-assessment/DevOps/elasticsearch\_helm/elasticsearch\_helm$ helm list



asraful@cks-master:~$ kubectl get pods



**values.yaml:**

global:

name: customer-abc-elasticsearch

namespace: default

esVersion: 7

esURI: http://customer-abc-headless:9200

deployment:

serviceName: customer-abc-elasticsearch-headless

clusterNodes: customer-abc-elasticsearch-0,customer-abc-elasticsearch-1,customer-abc-elasticsearch-2

containers:

images: elasticsearch:7.17.3

resources:

requests:

cpu: 0.2

memory: 100Mi

limits:

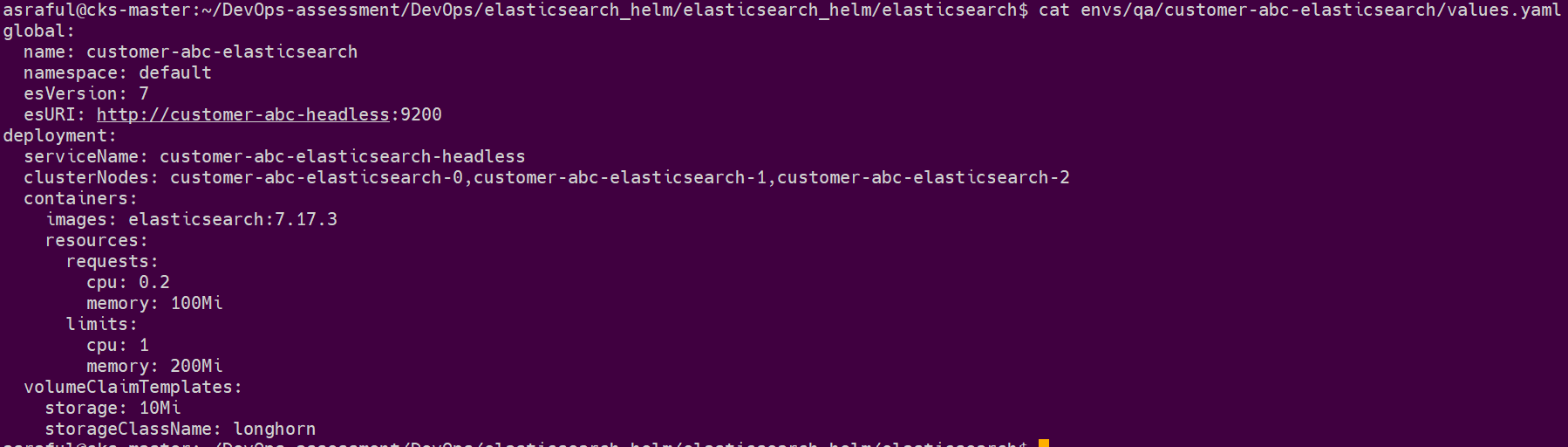
cpu: 1

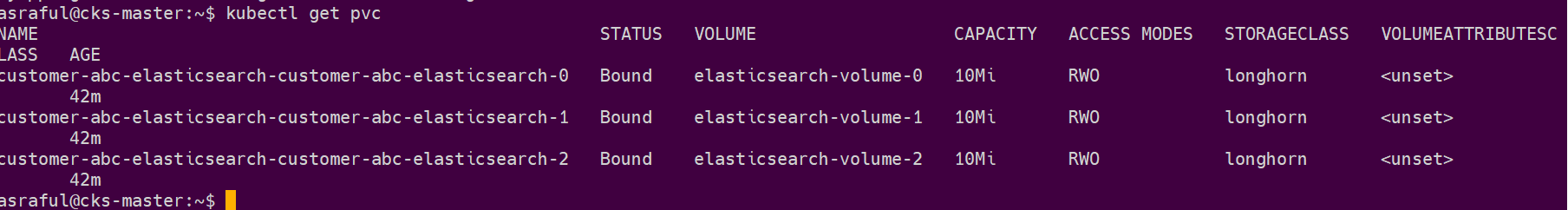
memory: 200Mi

volumeClaimTemplates:

storage: 10Mi

storageClassName: longhorn





# Metrics

1. Explain how Prometheus work.

* The Prometheus server queries(scrape) a list of data sources (sometimes called exporters) at a specific polling frequency.
* Prometheus data is stored in the form of metrics, with each metric having a name that is used for referencing and querying it.
* Prometheus stores data locally on disk, which helps for fast data storage and fast querying but ability to store metrics in remote storage.
* Each Prometheus server is standalone, not depending on network storage or other remote services.

1. How do you create custom Prometheus alerts and alerting rules for Kubernetes monitoring? Provide an example alert rule and its configuration.

**Solution:**

asraful@cks-master:~$ mkdir prometheus

asraful@cks-master:~$ cd prometheus/

asraful@cks-master:~/prometheus$ vi custom-values.yaml

# custom-values.yaml

prometheus:

service:

type: NodePort

grafana:

service:

type: NodePort

alertmanager:

service:

type: NodePort

asraful@cks-master:~/prometheus$ helm repo add prometheus-community https://prometheus-community.github.io/helm-charts

"prometheus-community" has been added to your repositories

asraful@cks-master:~/prometheus$ helm upgrade --install kube-prometheus-stack prometheus-community/kube-prometheus-stack -f custom-values.yaml

Release "kube-prometheus-stack" does not exist. Installing it now.

NAME: kube-prometheus-stack

LAST DEPLOYED: Thu Sep 19 06:39:06 2024

NAMESPACE: default

STATUS: deployed

REVISION: 1

NOTES:

kube-prometheus-stack has been installed. Check its status by running:

kubectl --namespace default get pods -l "release=kube-prometheus-stack"

asraful@cks-master:~/prometheus$ kubectl -n default get pods -l "release=kube-prometheus-stack"

NAME READY STATUS RESTARTS AGE

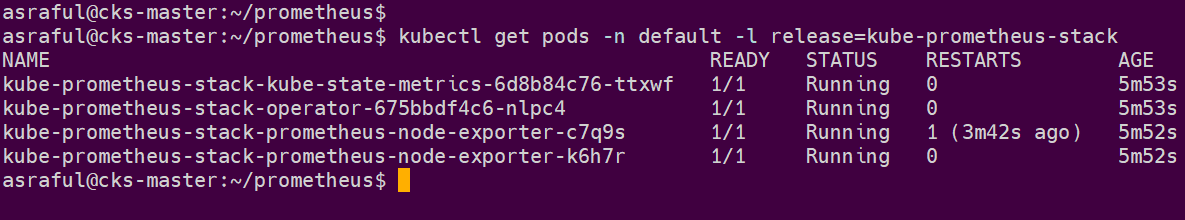
kube-prometheus-stack-kube-state-metrics-6d8b84c76-ttxwf 1/1 Running 0 4m9s

kube-prometheus-stack-operator-675bbdf4c6-nlpc4 1/1 Running 0 4m9s

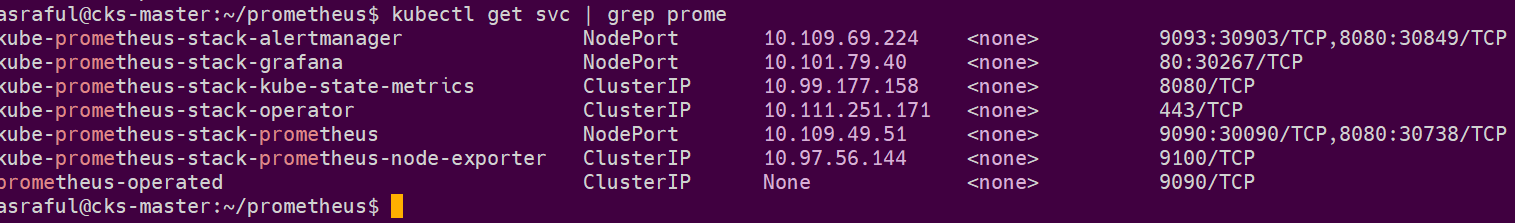
kube-prometheus-stack-prometheus-node-exporter-c7q9s 1/1 Running 1 (118s ago) 4m8s

kube-prometheus-stack-prometheus-node-exporter-k6h7r 1/1 Running 0 4m8s

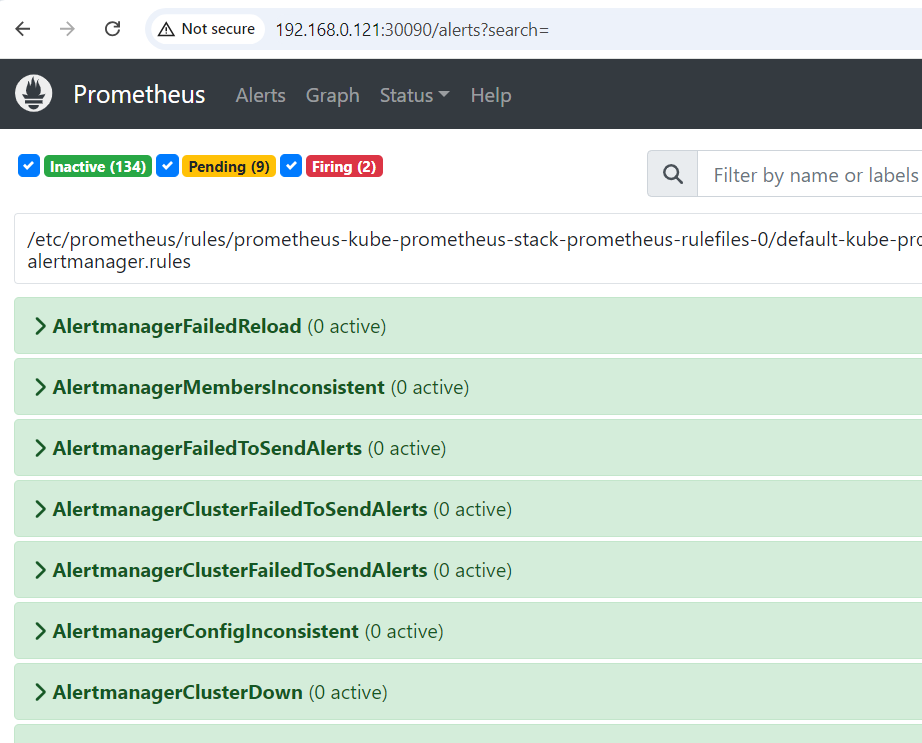
asraful@cks-master:~/prometheus$



asraful@cks-master:~/prometheus$ kubectl get svc | grep prome



http://192.168.0.121:30090/



asraful@cks-master:~/prometheus$ **vi custom-alert-rules.yaml**

apiVersion: monitoring.coreos.com/v1

kind: PrometheusRule

metadata:

labels:

app: kube-prometheus-stack

app.kubernetes.io/instance: kube-prometheus-stack

release: kube-prometheus-stack

name: kube-pod-not-ready

spec:

groups:

- name: my-pod-demo-rules

rules:

- alert: KubernetesPodNotHealthy

expr: sum by (namespace, pod) (kube\_pod\_status\_phase{phase=~"Pending|Unknown|Failed"}) > 0

for: 1m

labels:

severity: critical

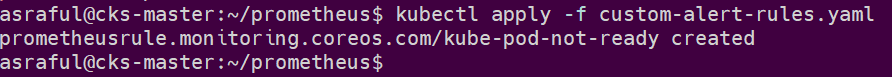
annotations:

summary: Kubernetes Pod not healthy (instance {{ $labels.instance }})

description: "Pod {{ $labels.namespace }}/{{ $labels.pod }} has been in a non-running state for longer than 15 minutes.\n VALUE = {{ $value }}\n LABELS = {{ $labels }}"

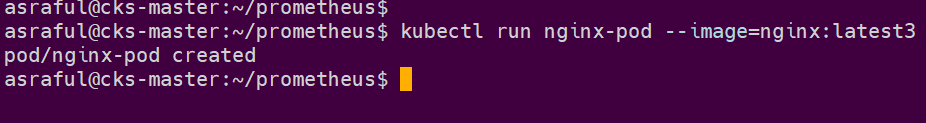
asraful@cks-master:~/prometheus$ kubectl apply -f custom-alert-rules.yaml

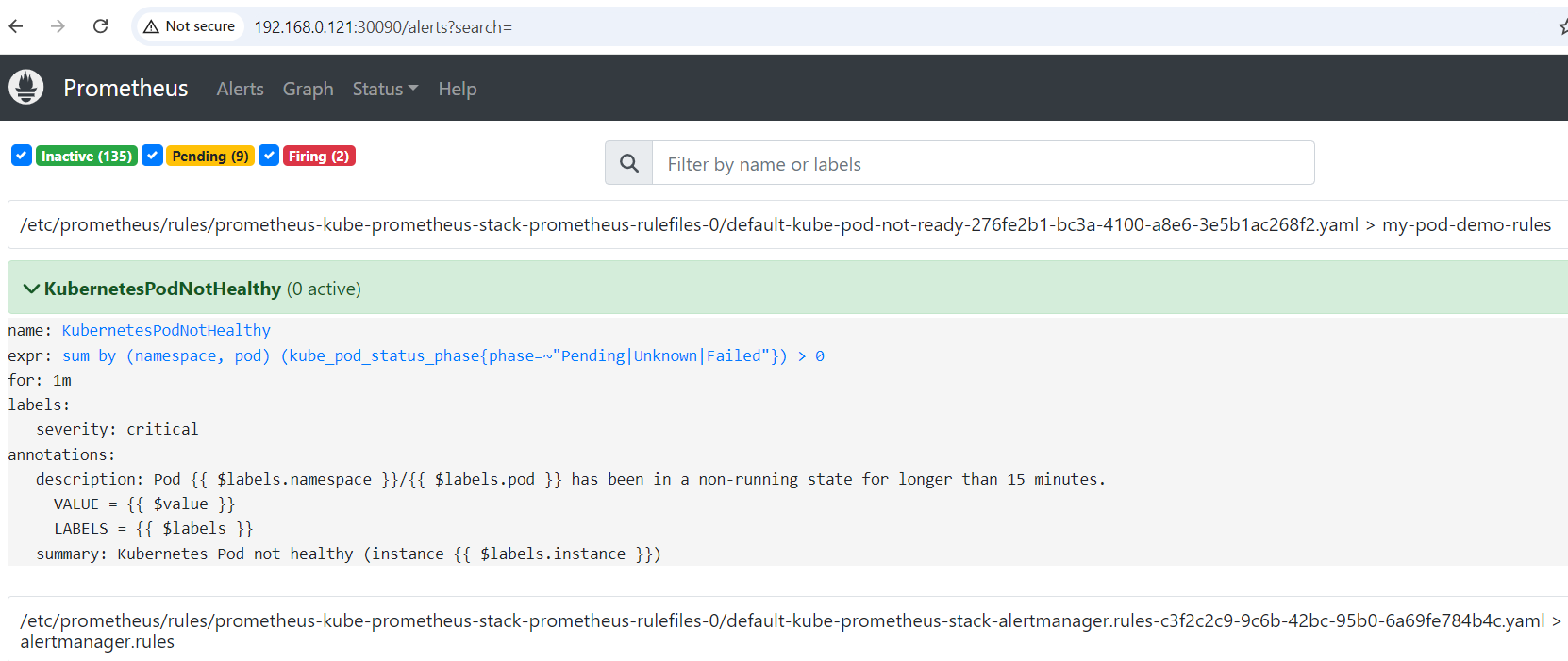
prometheusrule.monitoring.coreos.com/kube-pod-not-ready created

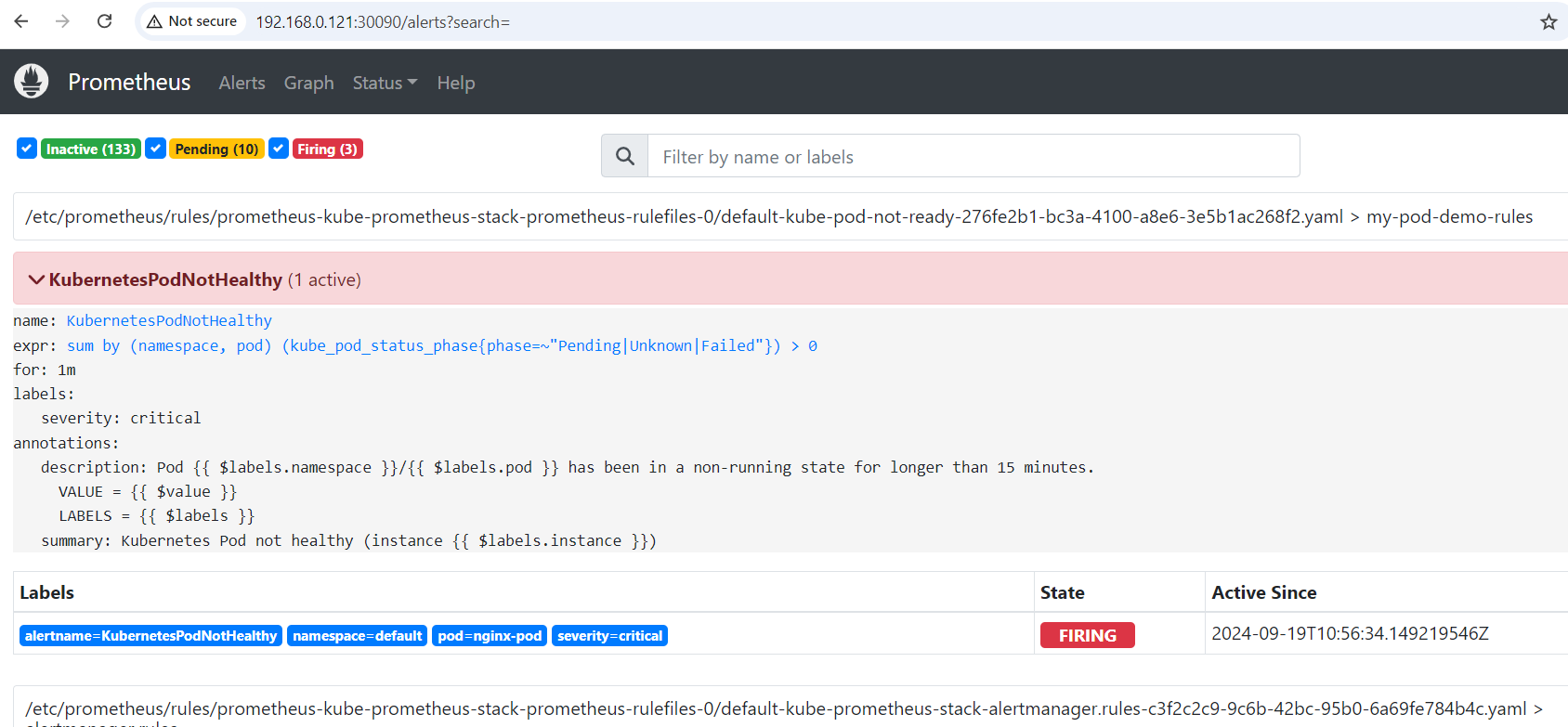


asraful@cks-master:~/prometheus$ kubectl run nginx-pod --image=nginx:latest3

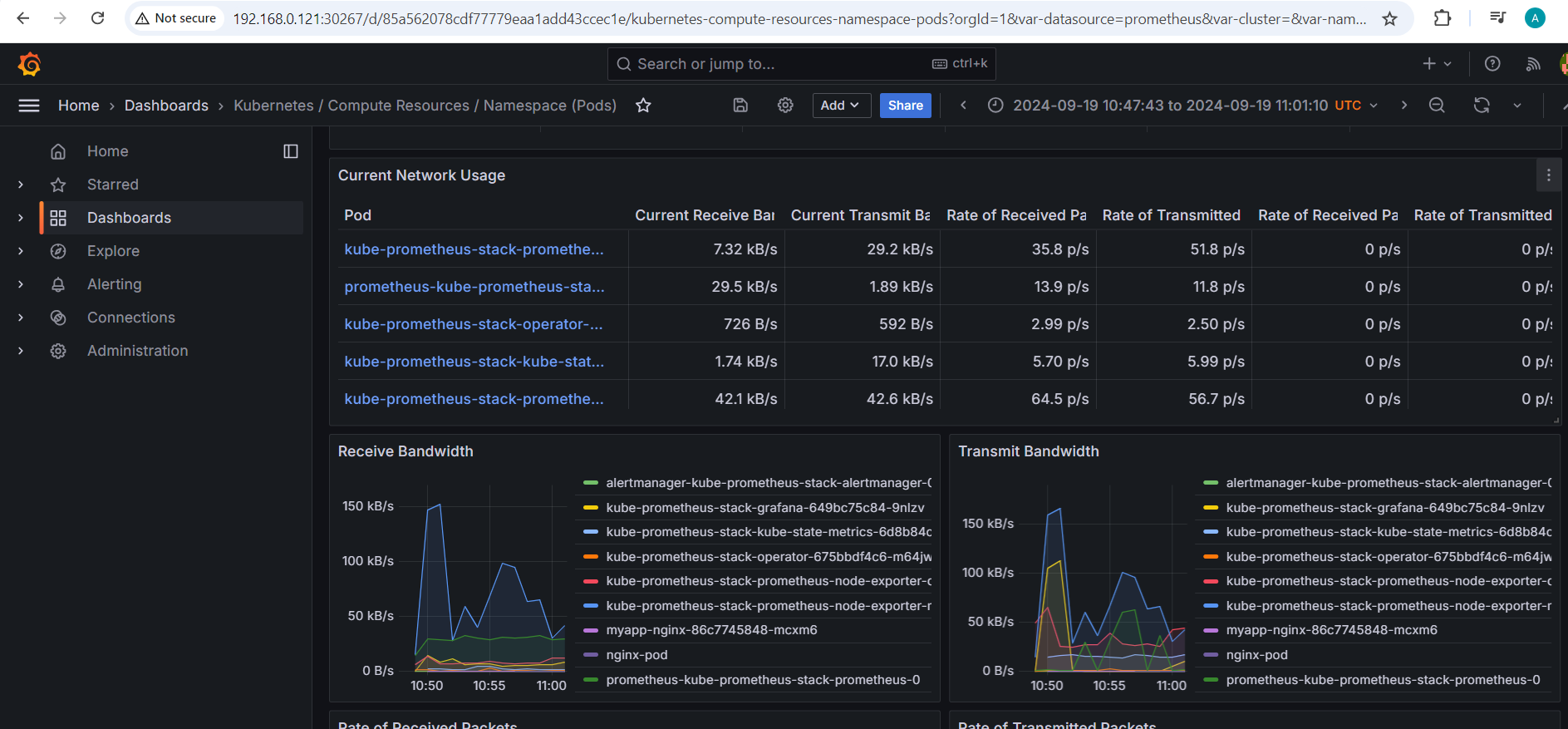
pod/nginx-pod created







http://192.168.0.121:30267/

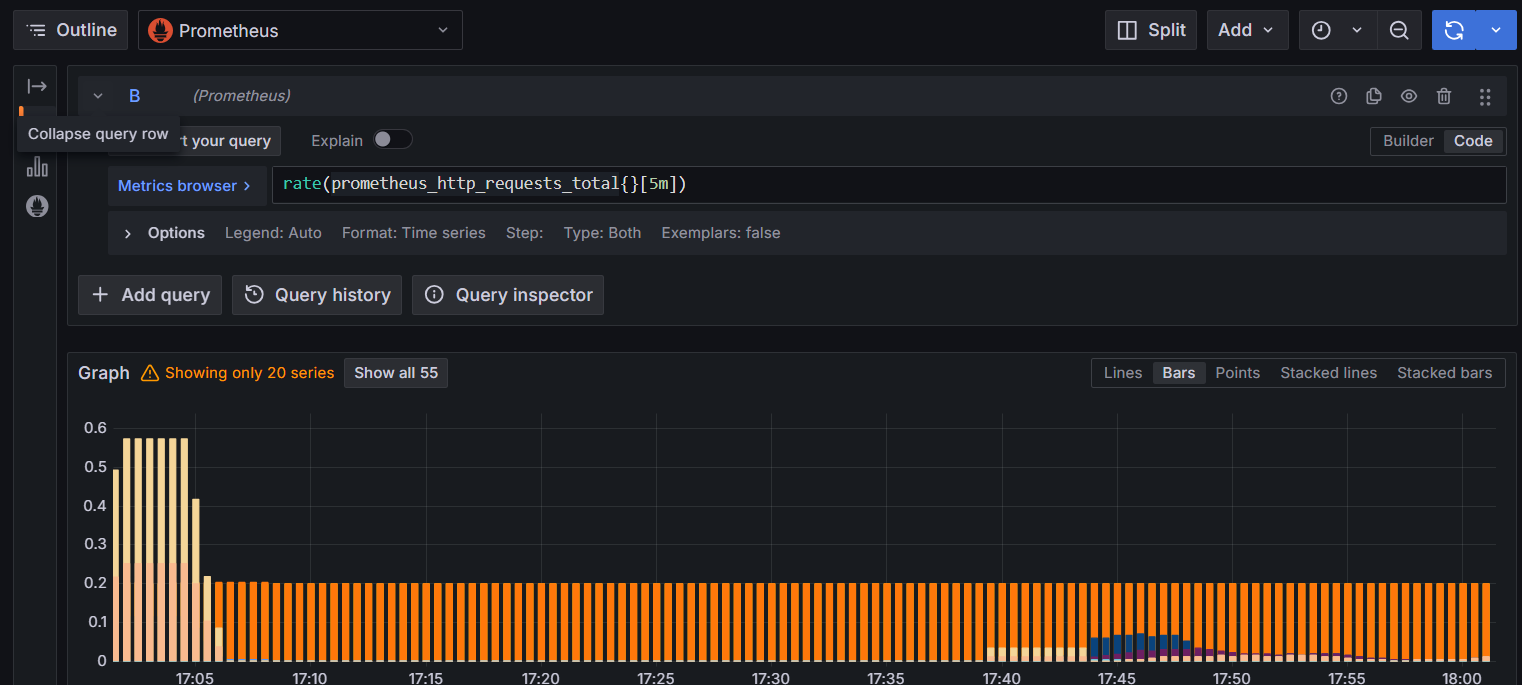


1. What is the Prometheus query you can use in Granfana to properly show usage trends of an application metric that is a counter?

**Solution:**

rate(http\_requests\_total{}[1m])

This query calculates the rate of http\_requests\_total over the last 1 minute.



Databases

Suggested environment: Cassandra 4.0 or above, mongo 4.4.0 or above

## Cassandra

Query to db cluster returns different result each time. Users reported query result has data records that they deleted days ago.

Explain what the likely reason for the behavior and how to avoid it.

**Possible reason:**

Data might not be fully synchronized across all nodes, leading to some nodes returning outdated information. If the read consistency level is set too low (e.g., ONE or TWO), you might get stale data from nodes that haven’t been updated yet.

**How to avoid it:**

Use a higher consistency level like QUORUM or ALL for reads to ensure that the data is consistent across a majority or all nodes.

Schedule regular nodetool repair operations to synchronize data across all nodes.

**nodetool repair**

## Mongo

We have mongodb replicaset\_1 with the following db and collections.



A sample record from company\_name:



Performance is bad as the hardware of replicaset\_1 is not capable to handle the database sanfrancisco. We added a new replicaset\_2.

Please provide all steps required to shard the collection sanfrancisco.company\_name based on \_id.

**Good Luck!**