#### 集群初始化命令

sudo kubeadm init \

--apiserver-advertise-address="192.168.9.132" \

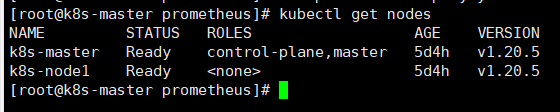
--image-repository="registry.aliyuncs.com/google\_containers" \

--kubernetes-version v1.20.5 \

--service-cidr="10.96.0.0/12" \

--pod-network-cidr="10.244.0.0/16" \

#### 查看集群状态



#### 配置prometheus

1、创建监控

kubectl create ns monitor-sa

2、部署 node-exporter

内容如下

apiVersion: apps/v1

kind: DaemonSet

metadata:

name: node-exporter

namespace: monitor-sa

labels:

name: node-exporter

spec:

selector:

matchLabels:

name: node-exporter

template:

metadata:

labels:

name: node-exporter

spec:

hostPID: true

hostIPC: true

hostNetwork: true

containers:

- name: node-exporter

image: prom/node-exporter:v0.16.0

ports:

- containerPort: 9100

resources:

requests:

cpu: 0.15

securityContext:

privileged: true

args:

- --path.procfs

- /host/proc

- --path.sysfs

- /host/sys

- --collector.filesystem.ignored-mount-points

- '"^/(sys|proc|dev|host|etc)($|/)"'

volumeMounts:

- name: dev

mountPath: /host/dev

- name: proc

mountPath: /host/proc

- name: sys

mountPath: /host/sys

- name: rootfs

mountPath: /rootfs

tolerations:

- key: "node-role.kubernetes.io/master"

operator: "Exists"

effect: "NoSchedule"

volumes:

- name: proc

hostPath:

path: /proc

- name: dev

hostPath:

path: /dev

- name: sys

hostPath:

path: /sys

- name: rootfs

hostPath:

path: /

#### 3、创建 sa 账号，对 sa 做 rbac 授权

#创建一个 sa 账号 monitor

kubectl create serviceaccount monitor -n monitor-sa

#把 sa 账号 monitor 通过 clusterrolebing 绑定到 clusterrole 上

kubectl create clusterrolebinding monitor-clusterrolebinding -n monitor-sa

#### 4、创建一个 configmap 存储卷，用来存放 prometheus 配置信息

具体内容为

kind: ConfigMap

apiVersion: v1

metadata:

labels:

app: prometheus

name: prometheus-config

namespace: monitor-sa

data:

prometheus.yml: |

global:

scrape\_interval: 15s

scrape\_timeout: 10s

evaluation\_interval: 1m

scrape\_configs:

- job\_name: 'kubernetes-node'

kubernetes\_sd\_configs:

- role: node

relabel\_configs:

- source\_labels: [\_\_address\_\_]

regex: '(.\*):10250'

replacement: '${1}:9100'

target\_label: \_\_address\_\_

action: replace

- action: labelmap

regex: \_\_meta\_kubernetes\_node\_label\_(.+)

- job\_name: 'kubernetes-node-cadvisor'

kubernetes\_sd\_configs:

- role: node

scheme: https

tls\_config:

ca\_file: /var/run/secrets/kubernetes.io/serviceaccount/ca.crt

bearer\_token\_file: /var/run/secrets/kubernetes.io/serviceaccount/token

relabel\_configs:

- action: labelmap

regex: \_\_meta\_kubernetes\_node\_label\_(.+)

- target\_label: \_\_address\_\_

replacement: kubernetes.default.svc:443

- source\_labels: [\_\_meta\_kubernetes\_node\_name]

regex: (.+)

target\_label: \_\_metrics\_path\_\_

replacement: /api/v1/nodes/${1}/proxy/metrics/cadvisor

- job\_name: 'kubernetes-apiserver'

kubernetes\_sd\_configs:

- role: endpoints

scheme: https

tls\_config:

ca\_file: /var/run/secrets/kubernetes.io/serviceaccount/ca.crt

bearer\_token\_file: /var/run/secrets/kubernetes.io/serviceaccount/token

relabel\_configs:

- source\_labels: [\_\_meta\_kubernetes\_namespace, \_\_meta\_kubernetes\_service\_name, \_\_meta\_kubernetes\_endpoint\_port\_name]

action: keep

regex: default;kubernetes;https

- job\_name: 'kubernetes-service-endpoints'

kubernetes\_sd\_configs:

- role: endpoints

relabel\_configs:

- source\_labels: [\_\_meta\_kubernetes\_service\_annotation\_prometheus\_io\_scrape]

action: keep

regex: true

- source\_labels: [\_\_meta\_kubernetes\_service\_annotation\_prometheus\_io\_scheme]

action: replace

target\_label: \_\_scheme\_\_

regex: (https?)

- source\_labels: [\_\_meta\_kubernetes\_service\_annotation\_prometheus\_io\_path]

action: replace

target\_label: \_\_metrics\_path\_\_

regex: (.+)

- source\_labels: [\_\_address\_\_, \_\_meta\_kubernetes\_service\_annotation\_prometheus\_io\_port]

action: replace

target\_label: \_\_address\_\_

regex: ([^:]+)(?::\d+)?;(\d+)

replacement: $1:$2

- action: labelmap

regex: \_\_meta\_kubernetes\_service\_label\_(.+)

- source\_labels: [\_\_meta\_kubernetes\_namespace]

action: replace target\_label: kubernetes\_namespace

- source\_labels: [\_\_meta\_kubernetes\_service\_name]

action: replace

target\_label: kubernetes\_name

#### 5、通过 deployment 部署 prometheus

apiVersion: apps/v1

kind: Deployment

metadata:

name: prometheus-server

namespace: monitor-sa

labels:

app: prometheus

spec:

replicas: 1

selector:

matchLabels:

app: prometheus

component: server

#matchExpressions:

#- {key: app, operator: In, values: [prometheus]}

#- {key: component, operator: In, values: [server]}

template:

metadata:

labels:

app: prometheus

component: server

annotations:

prometheus.io/scrape: 'false'

spec:

nodeName: k8s-node1

serviceAccountName: monitor

containers:

- name: prometheus

image: prom/prometheus:v2.2.1

imagePullPolicy: IfNotPresent

command:

- prometheus

- --config.file=/opt/prometheus/prometheus.yml

- --storage.tsdb.path=/prometheus

- --storage.tsdb.retention=720h

- --web.enable-lifecycle

ports:

- containerPort: 9090

protocol: TCP

volumeMounts:

- mountPath: /opt/prometheus/prometheus.yml

name: prometheus-config

subPath: prometheus.yml

- mountPath: /prometheus/

name: prometheus-storage-volume

volumes:

- name: prometheus-config

configMap:

name: prometheus-config

items:

- key: prometheus.yml

path: prometheus.yml

mode: 0644

- name: prometheus-storage-volume

hostPath:

path: /data

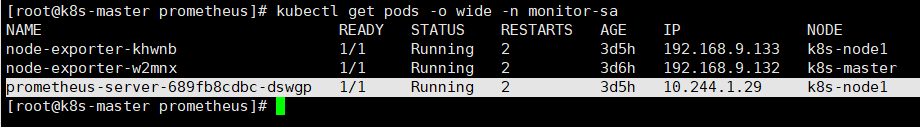
type: Directory

应用配置

kubectl apply -f prometheus-deploy.yaml

查看pods状态

kubectl get pods -o wide -n monitor-sa



6、给 prometheus pod 创建一个 service

apiVersion: v1

kind: Service

metadata:

name: prometheus

namespace: monitor-sa

labels:

app: prometheus

spec:

type: NodePort

ports:

- port: 9090

targetPort: 9090

protocol: TCP

nodePort: 31000

selector:

app: prometheus

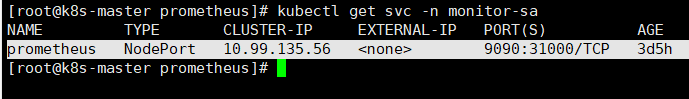
component: server

应用配置

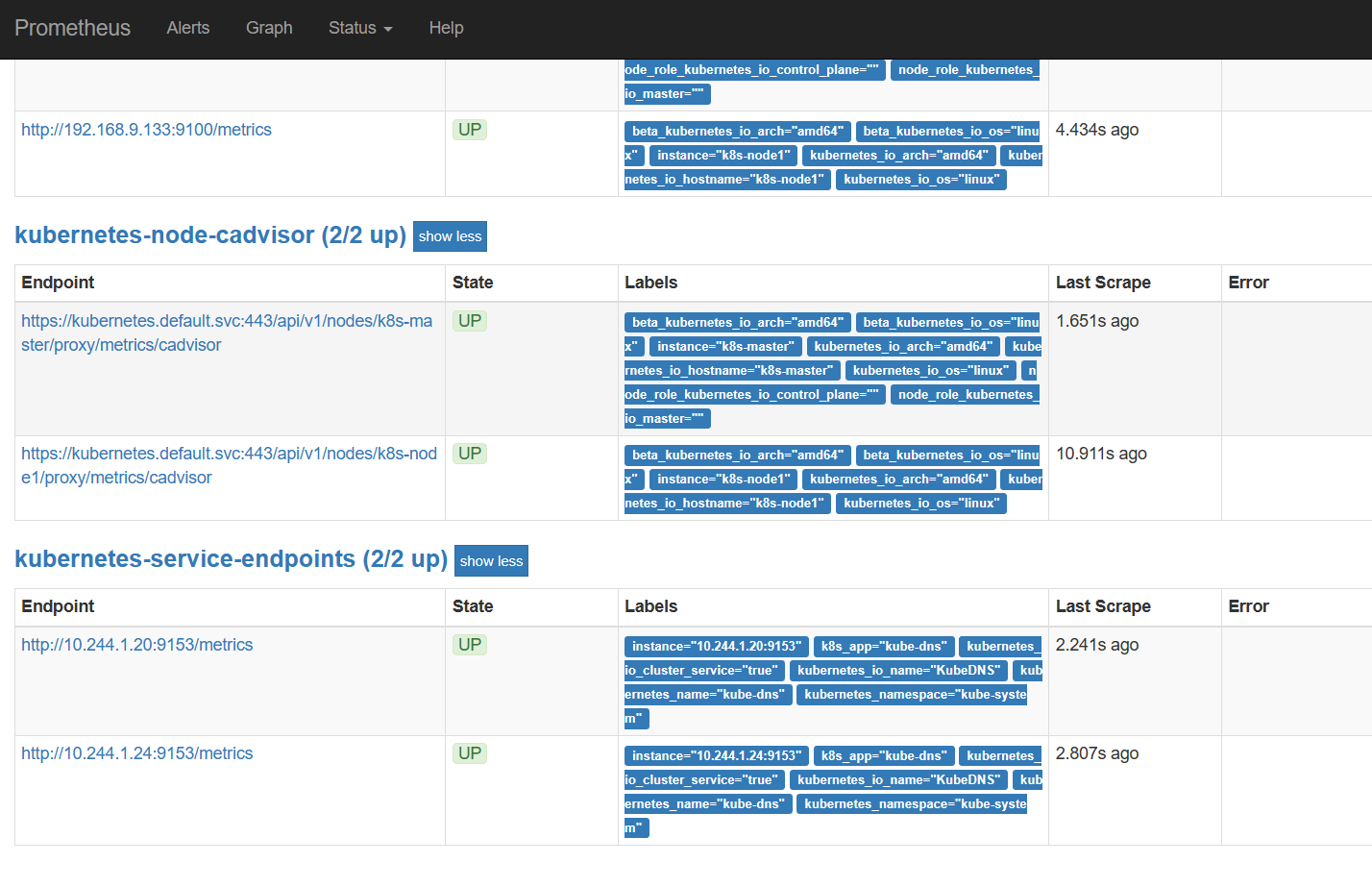
kubectl apply -f prometheus-svc.yaml

查看pods状态

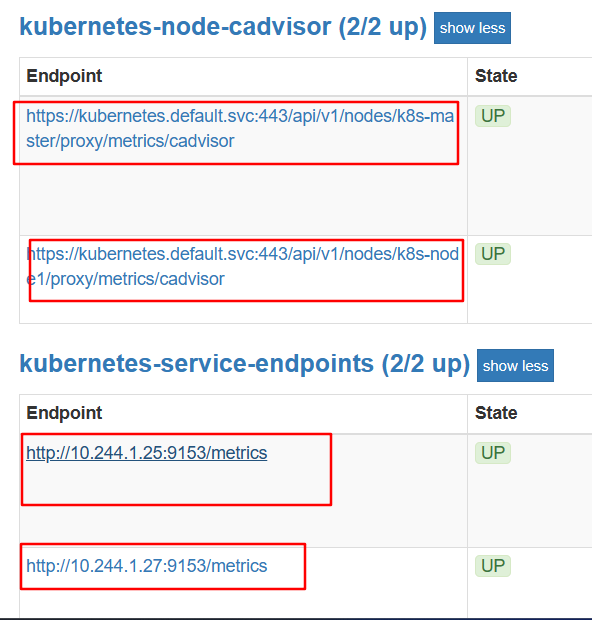
kubectl get svc -n monitor-sa



#### 7、浏览器访问页面查看targets



容器状态显示为UP但是当点击下面的IP地址时,网页显示错误,找不到数据



此时我在虚拟机向容器发送请求报错找不到路由,但是向虚拟机发送请求可以得到回应

