**Kubernetes部署基础nginx服务**

**一、编写nginx服务yaml文件**

**1.编写deployment.yaml文件**

|  |
| --- |
| Bash sudo mkdir /etc/nginx sudo nano /etc/nginx/nginx-deployment.yaml |

|  |
| --- |
| Bash #复制粘贴到文件中 apiVersion: apps/v1 kind: Deployment metadata:  name: nginx-deployment  namespace: nginx  labels:  app: nginx spec:  replicas: 1  selector:  matchLabels:  app: nginx  template:  metadata:  labels:  app: nginx  spec:  containers:  - name: nginx  image: nginx  ports:  - containerPort: 80 |

**2.编写service.yaml文件**

|  |
| --- |
| Bash sudo nano /etc/nginx-service.yaml |

|  |
| --- |
| Bash apiVersion: v1 kind: Service metadata:  name: nginx-service  labels:  app: nginx spec:  selector:  app: nginx  ports:  - protocol: TCP  port: 80  targetPort: 80  type: LoadBalancer |

**3.启动服务**

|  |
| --- |
| Bash #创建命名空间 kubectl create namespace nginx #创建ni=gnix pod kubectl create -f nginx-deployment.yaml kubectl create -f nginx-service.yaml |

**4.合并文件**

**1.模板-01**

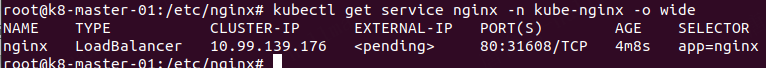
|  |
| --- |
| Bash cat << EOF > /etc/nginx/nginx-deployment.yaml --- apiVersion: apps/v1 kind: Deployment metadata:  name: nginx-deployment  namespace: nginx  labels:  app: nginx spec:  replicas: 2  selector:  matchLabels:  app: nginx  template:  metadata:  labels:  app: nginx  spec:  containers:  - name: nginx  image: registry.docker.io:5000/nginx  ports:  - containerPort: 80 --- apiVersion: v1 kind: Service metadata:  name: nginx-service  labels:  app: nginx spec:  selector:  app: nginx  ports:  - protocol: TCP  port: 80  targetPort: 80  type: LoadBalancer EOF |

**2.模板-02**

|  |
| --- |
| Bash #nginx\_deployment\_service.yaml cat << EOF > /etc/nginx/nginx-deployment.yaml apiVersion: v1 kind: Namespace metadata:  name: kube-nginx --- apiVersion: apps/v1 kind: Deployment metadata:  name: nginx  namespace: kube-nginx spec:  selector:  matchLabels:  app: nginx  template:  metadata:  labels:  app: nginx  spec:  containers:  - name: nginx  image: registry.docker.io:5000/nginx  ports:  - name: http  containerPort: 80   --- apiVersion: v1 kind: Service metadata:  namespace: metallb-system  name: nginx spec:  ports:  - name: http  port: 80  protocol: TCP  targetPort: 80  selector:  app: nginx  type: LoadBalancer EOF |

**5.查看服务是否对外暴露**

|  |
| --- |
| Bash kubectl get service nginx -n kube-nginx -o wide |



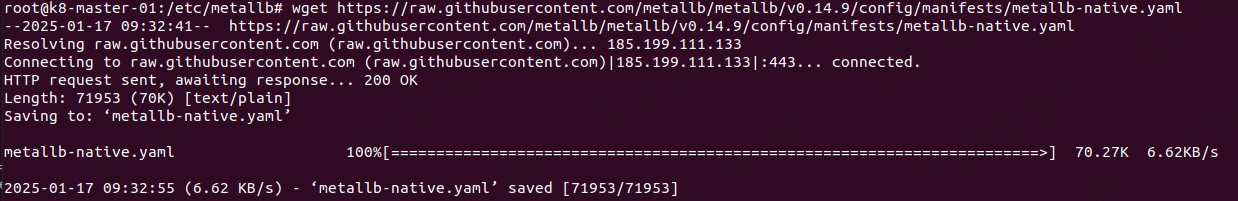
上面没有外部IP因为缺少两个组件以及没有配置地址池

**二、下载METALLB**

**1.下载配置文件**

|  |
| --- |
| Bash #使用metallb插件 #参考官网 https://metallb.io/installation/ |

|  |
| --- |
| Bash sudo mkdir /etc/metallb cd /etc/metallb wget https://raw.githubusercontent.com/metallb/metallb/v0.14.9/config/manifests/metallb-native.yaml |



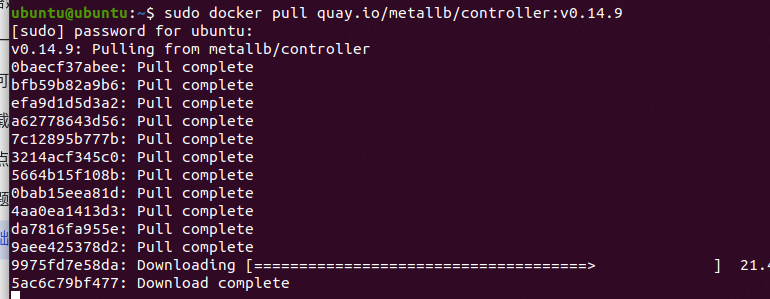
**2.查看配置文件镜像**

|  |
| --- |
| Bash #查看文件所需镜像 grep image /etc/metallb/metallb-native.yaml |



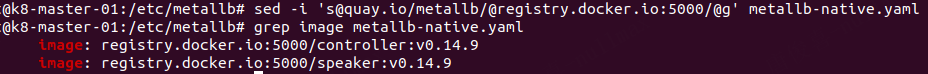
**3.下载镜像并打标签上传到私有仓库**

|  |
| --- |
| Bash #使用docker下载镜像并上传到私服仓库 sudo docker pull quay.io/metallb/controller:v0.14.9 sudo docker tag quay.io/metallb/controller:v0.14.9 registry.docker.io:5000/controller:v0.14.9 sudo docker push registry.docker.io:5000/controller:v0.14.9 |



**4.修改镜像**

|  |
| --- |
| Bash #修改镜像 sed -i 's@quay.io/metallb/@registry.docker.io:5000/@g' /etc/metallb/metallb-native.yaml |

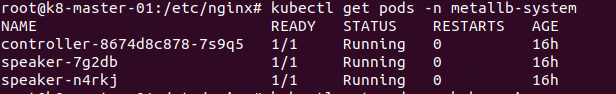


**5.启用metallb**

|  |
| --- |
| Bash kubectl create -f /etc/metallb/metallb-native.yaml |

**6.查看metallb是否正常工作**

|  |
| --- |
| Bash kubectl get pods -n metallb-system |



**7.配置暴露服务文件**

|  |
| --- |
| Bash # #nginx\_deployment\_service.yaml cat << EOF > /etc/nginx/nginx-deployment.yaml apiVersion: metallb.io/v1beta1 kind: IPAddressPool metadata:  name: ip-pool-01  namespace: metallb-system spec:  addresses:  - 192.168.100.10-192.168.100.250 --- apiVersion: metallb.io/v1beta1 kind: L2Advertisement metadata:  name: level2-advertisement  namespace: metallb-system spec:  ipAddressPools:  - ip-pool-01 # nodeSelectors: # - matchLabels: # kubernetes.io/hostname: k8-slave-01  EOF |

**8.再起重启nginx服务**

|  |
| --- |
| Bash kubectl delete -f /etc/nginx/nginx-delployment.yaml kubectl create -f /etc/nginx/nginx-delployment.yaml |



**9.访问宿主机IP：**

