安装、搭建

安装

ubuntu安装

> https://developer.aliyun.com/article/906571
>
> Ubuntu 18.04 搭建单节点 k8s 记录 (不需要科学上网的方案) | Server 运维论坛 (learnku.com)

- 环境配置

```
swapoff -a

gedit /etc/fstab
# 注释以下行：
/swapfile none swap sw 0 0
```

- 安装

```
# 追加k8s的包地址
echo "deb https://mirrors.aliyun.com/kubernetes/apt kubernetes-xenial main" >> /etc/apt/sources.list
# 添加key
curl https://mirrors.aliyun.com/kubernetes/apt/doc/apt-key.gpg | sudo apt-key add
# 更新apt
apt-get update

apt-get install -y apt-transport-https curl
apt-get install -y kubelet kubeadm kubectl --allow-unauthenticated
```

搭建

初始化kubeadm

```
kubeadm init \
  --apiserver-advertise-address=192.168.1.88 \
  --image-repository registry.aliyuncs.com/google_containers \
  --kubernetes-version v1.17.17 \
  --service-cidr=10.1.0.0/16 \
  --pod-network-cidr=10.244.0.0/16
## --pod-network-cidr：pod的网络地址空间

# 配置结点
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

安装需要的镜像

```
kubeadm config images list
```

例如得到：

```
registry.k8s.io/kube-apiserver:v1.27.2
registry.k8s.io/kube-controller-manager:v1.27.2
registry.k8s.io/kube-scheduler:v1.27.2
registry.k8s.io/kube-proxy:v1.27.2
registry.k8s.io/pause:3.9
registry.k8s.io/etcd:3.5.7-0
registry.k8s.io/coredns/coredns:v2.0.0-rc7
```

一键脚本：

```
for  i  in  `kubeadm config images list`;  do
    imageName=${i#k8s.gcr.io/}
    docker pull registry.aliyuncs.com/google_containers/$imageName
    docker tag registry.aliyuncs.com/google_containers/$imageName
k8s.gcr.io/$imageName
    docker rmi registry.aliyuncs.com/google_containers/$imageName
done;
```

安装dashboard

> Kubernetes基础概念 (yuque.com)

```
kubectl apply -f
https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.0-
rc7/aio/deploy/recommended.yaml

docker pull registry.aliyuncs.com/google_containers/kubernetes-dashboard-
amd64:v1.10.1
docker tag registry.aliyuncs.com/google_containers/kubernetes-dashboard-
amd64:v1.10.1 k8s.gcr.io/kubernetes-dashboard-amd64:v1.10.1
docker rmi registry.aliyuncs.com/google_containers/kubernetes-dashboard-
amd64:v1.10.1

kubectl create -f kubernetes-dashboard.yaml

# 查看安装结果
kubectl get pod --namespace=kube-system

# 改type为NodePort
kubectl edit svc kubernetes-dashboard -n kubernetes-dashboard

# 找到端口，在安全组放行
kubectl get svc -A |grep kubernetes-dashboard

# 创建访问账号，将配置内容写入dashboard-admin.yaml
apiVersion: v1
kind: ServiceAccount
```

```
metadata:
  name: admin-user
  namespace: kubernetes-dashboard
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: admin-user
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: admin-user
  namespace: kubernetes-dashboard

# 安装
kubectl apply -f dashboard-admin.yaml

#获取访问令牌
kubectl -n kubernetes-dashboard get secret $(kubectl -n kubernetes-dashboard get sa/admin-user -o jsonpath="{.secrets[0].name}") -o go-template="{{.data.token | base64decode}}"
```

生效主节点

```
# 添加环境变量
echo "export KUBECONFIG=/etc/kubernetes/admin.conf" >> ~/.bashrc
# 添加网络插件
kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
# 保存
kubeadm join

# 单节点使用：让master也参与调度
kubectl taint nodes --all node-role.kubernetes.io/master-
```