无锡城市职业技术学院

**实 验（实 训）报 告**

**所属课程名称**  云平台构建与运维

**班 级** 计网2210

**学 号**  1422040312

**姓 名**  杨雪

**指导教师**  徐峰

无锡城市职业技术学院教务处监制

**项目名称：Kubernetes安装和使用**

日期：2024年5月5日

|  |
| --- |
| **一、实验（实训）概述：** |
| **【目的及要求】**  学习部署原生Kubernettes云平台以及Kubectl运行容器的使用  **【实验环境】**  虚拟机K8s master,虚拟机k8s node |
| **二、实验（实训）内容：** |
| **【实验（实训）过程】（步骤、记录、数据、程序等）**  **原生Kubernetes云平台部署**  **1. 节点规划**  Kubernetes 集 群 各 节 点 的 规 划 见 表所 示 ， 各 节 点 内 存 均 为 8GB ， 使 用CentOS-7-x86\_64-DVD-1804.iso 安装操作系统。   |  |  |  | | --- | --- | --- | | **IP 地址** | **主机名** | **节点** | | 192.168.100.143 | master | master 节点 | | 192.168.100.130 | node | node 节点 |   **2. 配置 hosts**  两个节点配置相同，以 master 节点为例。 （nodo也要配置）  [root@master ~]# vi /etc/hosts  192.168.100.143 master  192.168.100.130 node      **3. 停止防火墙和 SELinux**  在 master 和 node 节点进行以下配置，以 master 节点为例。  1）停止防火墙  [root@master ~]# systemctl stop firewalld  [root@master ~]# systemctl disable firewalld    2）停止 SELinux  [root@master ~]# setenforce 0  [root@master ~]# vi /etc/selinux/config  SELINUX=permissive      **4. 关闭 Swap**  在 master 和 node 节点进行以下配置，以 master 节点为例。  1）停用 Swap  [root@master ~]# swapoff -a    2）删除 fstab 中的 SWAP 挂载  [root@master ~]# vi /etc/fstab  删除：  /dev/mapper/centos-swap swap swap defaults 0 0      **删除**  **5. 配置 YUM**  在 master 进行以下配置：  1）将 CentOS-7-x86\_64-DVD-1804.iso 和 K8S.tar.gz 通过 SFTP 传输到 master 节点的/root 目录。    **直接拖进去**  [root@master ~]# ls  anaconda-ks.cfg CentOS-7-x86\_64-DVD-1804.iso K8S.tar.gz    2）创建挂载目录。  [root@master ~]# mkdir /opt/centos    3）编辑 fstab。  [root@master ~]# vi /etc/fstab    在最下面添加一行：  /root/CentOS-7-x86\_64-DVD-1804.iso /opt/centos iso9660  defaults,ro,loop 0 0    4）挂载 ISO。  [root@master ~]# mount -a  [root@master ~]# mount  ……    /root/CentOS-7-x86\_64-DVD-1804.iso on /opt/centos type iso9660 (ro,relatime)    5）解压 K8S。  [root@master ~]# mkdir /opt/K8S    [root@master ~]# tar -zxf K8S.tar.gz -C /opt/K8S    6）编辑 YUM 配置文件。  [root@master ~]# cd /etc/yum.repos.d/  [root@master yum.repos.d]# rm -f \*  [root@master yum.repos.d]# vi local.repo    [centos]  name=centos  baseurl=file:///opt/centos  gpgcheck=0  [K8S]  name=K8S  baseurl=file:///opt/K8S/Kubernetes  gpgcheck=0    7）清空缓存。  [root@master ~]# yum clean all    8）生成新缓存。  [root@master ~]# yum makecache    9）安装 vsftpd。  [root@master ~]# yum -y install vsftpd    10）编辑 vsftpd 配置文件。  [root@master ~]# vi /etc/vsftpd/vsftpd.conf    增加一行：  anon\_root=/opt    11）启动和启用服务。  [root@master ~]# systemctl start vsftpd  [root@master ~]# systemctl enable vsftpd    12）在 node 节点编辑 YUM 配置文件。  [root@node ~]# cd /etc/yum.repos.d/  [root@node yum.repos.d]# rm -f \*  [root@node yum.repos.d]# vi ftp.repo    [centos]  name=centos  baseurl=ftp://192.168.100.60/centos  gpgcheck=0  [K8S]  name=K8S  baseurl=ftp://192.168.100.60/K8S/Kubernetes  gpgcheck=0    13）在 node 节点清空缓存。  [root@node ~]# yum clean all    14）在 node 节点生成新缓存。  [root@node ~]# yum makecache    **6. 配置 NTP**  1）在 master 节点上安装 chrony。  [root@master ~]# yum -y install chrony    2）编辑配置文件。  master 节点修改/etc/chrony.conf 文件，删除默认 NTP 服务器，指定上游公共 NTP 服务  器，并允许其他节点同步时间。  [root@master ~]# vi /etc/chrony.conf  删除 4 个 server，添加以下配置：  local stratum 10  server master iburst  allow all    **删除**    **添加**  3）启动服务及启用服务。  [root@master ~]# systemctl start chronyd  [root@master ~]# systemctl enable chronyd    4）查看时间同步源。  [root@master ~]# chronyc sources -v  MS Name/IP address Stratum Poll Reach LastRx Last sample  ==================================================================  =============  ^\* master 10 6 77 8 -1ns[-8190ns] +/- 20us    5）在 node 节点上安装 chrony。  [root@node ~]# yum -y install chrony    6）编辑配置文件。  node 节点修改/etc/chrony.conf 文件，指定内部 master 节点为上游 NTP 服务器。  [root@node ~]# vi /etc/chrony.conf    删除 4 个 server，添加以下配置：  server master iburst    **添加**  **删除**  7）启动服务及启用服务。  [root@node ~]# systemctl start chronyd  [root@node ~]# systemctl enable chronyd     1. 查看时间同步源。   [root@node ~]# chronyc sources  210 Number of sources = 1  MS Name/IP address Stratum Poll Reach LastRx Last  sample  ======================================================================  ========  ^\* master 3 6 17 40 -5130ns[ -73us] +/- 43ms  查询结果中如果存在以“^\*”开头的行，即说明已经同步成功。    **7. 配置路由转发**  RHEL7/CentOS7上的一些用户报告了由于iptables被绕过而导致流量路由不正确的问题，  所以需要在各节点开启路由转发。  在 master 和 node 节点进行以下配置，以 master 节点为例。  1）创建/etc/sysctl.d/k8s.conf。  [root@master ~]# vi /etc/sysctl.d/k8s.conf    net.ipv4.ip\_forward = 1  net.bridge.bridge-nf-call-ip6tables = 1  net.bridge.bridge-nf-call-iptables = 1    2）载入内核模块。  [root@master ~]# modprobe br\_netfilter    3）使配置生效。  [root@master ~]# sysctl -p /etc/sysctl.d/k8s.conf    **8. 配置 IPVS**  由于 IPVS 已经加入到了内核的主干，所以为 kube-proxy 开启 IPVS 的前提需要加载以  下的内核模块。在所有节点执行以下操作。  在 master 和 node 节点进行以下配置，以 master 节点为例。  1）编辑文件。  [root@master ~]# vi /etc/sysconfig/modules/ipvs.modules    #!/bin/bash  modprobe -- ip\_vs  modprobe -- ip\_vs\_rr  modprobe -- ip\_vs\_wrr  modprobe -- ip\_vs\_sh  modprobe -- nf\_conntrack\_ipv4    2）增加执行权限。  [root@master ~]# chmod 755 /etc/sysconfig/modules/ipvs.modules    3）执行脚本。  [root@master ~]# bash /etc/sysconfig/modules/ipvs.modules    4）显示已载入的模块。  [root@master ~]# lsmod | grep -e ip\_vs -e nf\_conntrack\_ipv4  nf\_conntrack\_ipv4 15053 0  nf\_defrag\_ipv4 12729 1 nf\_conntrack\_ipv4  ip\_vs\_sh 12688 0 ip\_vs\_wrr 12697 0  ip\_vs\_rr 12600 0  ip\_vs 141432 6 ip\_vs\_rr,ip\_vs\_sh,ip\_vs\_wrr  nf\_conntrack 133053 2 ip\_vs,nf\_conntrack\_ipv4  libcrc32c 12644 3 xfs,ip\_vs,nf\_conntrack    5）安装 ipset 软件包。  [root@master ~]# yum -y install ipset ipvsadm      **9. 安装 Docker**  Kubernetes 默认的容器运行时仍然是 Docker，使用的是 Kubelet 中内置 dockershim CRI  实现。需要注意的是，这里统一使用 Docker18.09 版本。  在 master 和 node 节点进行以下配置，以 master 节点为例。  1）安装 yum-utils。  [root@master ~]# yum -y install yum-utils      2）安装支持软件。  [root@master ~]# yum install -y device-mapper-persistent-data lvm2      3）安装 Docker。  [root@master ~]# yum -y install docker-ce-18.09.6 docker-ce-cli-18.09.6 containerd.io      4）创建目录。  [root@master ~]# mkdir /etc/docker    5）编辑 Docker 配置文件。  [root@master ~]# vi /etc/docker/daemon.json    {  "exec-opts": ["native.cgroupdriver=systemd"]  }    6）启动及启用 Docker 服务。  [root@master ~]# systemctl daemon-reload  [root@master ~]# systemctl start docker  [root@master ~]# systemctl enable docker    7）查看 docker info。  [root@master ~]# docker info | grep Cgroup  Cgroup Driver: systemd    **10. 安装 Kubeadm 工具**  Kubelet 负责与其他节点集群通信，并进行本节点 Pod 和容器生命周期的管理。Kubeadm  是 Kubernetes 的自动化部署工具，降低了部署难度，提高效率。Kubectl 是 Kubernetes 集群  管理工具。  在 master 和 node 节点进行以下配置，以 master 节点为例。  1）安装软件。  [root@master ~]# yum -y install kubelet-1.14.1 kubeadm-1.14.1 kubectl-1.14.1    2）启动和启用 kubelet。  [root@master ~]# systemctl enable kubelet  [root@master ~]# systemctl start kubelet    **11. 初始化 Kubernetes 集群在 master 节点进行以下操作。**  1）加载镜像。  [root@master ~]# cd /opt/K8S  [root@master K8S]# ./kubernetes\_base.sh    2）初始化 Kubernetes 集群。  [root@master ~]#  kubeadm init  --apiserver-advertise-address 192.168.100.60  --kubernetes-version="v1.14.1"  --pod-network-cidr=10.16.0.0/16  --image-repository=registry.aliyuncs.com/google\_containers    ……  Your Kubernetes control-plane has initialized successfully!  ……  Then you can join any number of worker nodes by running the following on each as root:  kubeadm join 192.168.100.60:6443 --token tn8fi0.32ubkm8u987igkbo \  --discovery-token-ca-cert-hash  sha256:897afbc11a85991f995d0d8b007b152460a94478a554cb89d895b99bf835f041    3）后续配置。  Kubectl默认会在执行的用户home目录下面的.kube目录下寻找config文件，配置kubectl  工具。  [root@master ~]# mkdir -p $HOME/.kube  [root@master ~]# sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config  [root@master ~]# sudo chown $(id -u):$(id -g) $HOME/.kube/config    4）检查集群状态。  [root@master ~]# kubectl get cs  NAME STATUS MESSAGE ERROR  scheduler Healthy ok  controller-manager Healthy ok  etcd-0 Healthy {"health":"true"}    **12. 配置 Kubernetes 网络**  在 master 节点部署 flannel 网络，使用 kubectl apply 命令安装网络。  1）进入 yaml 目录。  [root@master ~]# cd /opt/K8S/yaml    2）部署 flannel 网络。  [root@master yaml]# kubectl apply -f kube-flannel.yaml    3）查看状态。  [root@master ~]# kubectl get pods -n kube-system  NAME READY STATUS RESTARTS AGE  coredns-8686dcc4fd-mfsln 1/1 Running 0 6m35s  coredns-8686dcc4fd-nnjk7 1/1 Running 0 6m35s  etcd-master 1/1 Running 0 5m47s  kube-apiserver-master 1/1 Running 0 5m41s  kube-controller-manager-master 1/1 Running 0 5m55s  kube-flannel-ds-amd64-lfvp8 1/1 Running 0 42skube-proxy-kcdcx 1/1 Running 0 6m35s  kube-scheduler-master 1/1 Running 0 5m32s    **13. 将 node 节点加入集群**  1）将 K8S.tar.gz 通过 SFTP 传输到 node 节点的/root 目录。    **直接拖进来**  [root@node ~]# ls  anaconda-ks.cfg K8S.tar.gz    2）解压 K8S。  [root@node ~]# tar -zxf K8S.tar.gz    3）加载镜像。  [root@node ~]# ls  anaconda-ks.cfg images K8S.tar.gz Kubernetes kubernetes\_base.sh yaml    [root@node ~]# ./kubernetes\_base.sh    4）在 node 节点执行之前初始化 Kubernetes 集群时最后提供的加入节点的命令。  [root@node ~]# kubeadm join 192.168.100.60:6443 --token tn8fi0.32ubkm8u987igkbo \  >  --discovery-token-ca-cert-hash  sha256:897afbc11a85991f995d0d8b007b152460a94478a554cb89d895b99bf835f041  ……  This node has joined the cluster:  ……    5）如果没有记下当时给出的提示，可以在 master 节点执行以下命令重新生成 Token 并  查看。  [root@master ~]# kubeadm token create --print-join-command  kubeadm join 192.168.100.60:6443  --token ldcvqa.ijb0u7jyj8rwv5w0  --discovery-token-ca-cert-hash  sha256:897afbc11a85991f995d0d8b007b152460a94478a554cb89d895b99bf835f041    6）在 master 节点检查各节点状态。  [root@master ~]# kubectl get nodes  NAME STATUS ROLES AGE VERSION  master Ready master 6m40s v1.14.1  node Ready <none> 48s v1.14.1    **14. 安装 Dashboard**  1）进入/opt/K8S/yaml 目录，使用 kubectl create 命令安装 Dashboard。  [root@master ~]# cd /opt/K8S/yaml  [root@master yaml]# kubectl create -f kubernetes-dashboard.yaml  [root@master yaml]# kubectl create -f dashboard-adminuser.yaml    2）检查所有 Pod 状态。  [root@master ~]# kubectl get pods --all-namespaces -o wide  NAMESPACE NAME READY STATUS RESTARTS AGE IP NODE NOMINATED  NODE READINESS GATES  kube-system kubernetes-dashboard-5f7b999d65-g9qv5 1/1 Running 0 93s  10.16.1.2 node <none> <none>    3）通过命令检查到 kubernetes-dashboard 被调度到 Node 节点运行，通过 Firefox 浏览器中输入 Node 节点地址（Master 也可以访问）“https://192.168.100.61:30000”，即可访问 Kubernetes Dashboard，    **使用kubectl运行容器**  **1. 载入或下载镜像**  1）在 node 节点载入 nginx:latest 镜像。  [root@node ~]# ls  anaconda-ks.cfg nginx\_latest.tar  [root@node ~]# docker load -i nginx\_latest.tar  2）或从 Docker Hub 下载镜像。  [root@node ~]# docker pull nginx:latest    3）查看镜像。  [root@node ~]# docker images | grep nginx  nginx latest 540a289bab6c 2 months ago 126MB    **2. 使用 kubectl 运行容器**  1）创建 deployment。  [root@master ~]# kubectl create deployment nginx --image=nginx  deployment.apps/nginx created    2）查看 Pods，验证 Pods 是否正常运行。  [root@master ~]# kubectl get pods  NAME READY STATUS RESTARTS AGE  nginx-65f88748fd-wf62w 1/1 Running 0 12s  可以看到容器在运行中。    3）查看所有 deployment。  [root@master ~]# kubectl get deployment  NAME READY UP-TO-DATE AVAILABLE AGE  nginx 1/1 1 1 35s    4）采用 NodePort 的方式来暴露 nginx 服务。  [root@master ~]# kubectl expose deployment nginx --port=80 --type=NodePortservice/nginx exposed    5）查看 Service。  [root@master ~]# kubectl get svc  NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE  kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 23h  nginx NodePort 10.100.233.224 <none> 80:31507/TCP 19s     1. 在本机通过浏览器访问 Nginx 应用     7）pod 动态伸缩。  运行以下命令，将容器数量更新为 3 个。  [root@master ~]# kubectl scale deployment nginx --replicas=3  deployment.extensions/nginx scaled    8）查看 pod 的容器数量。  [root@master ~]# kubectl get pods  NAME READY STATUS RESTARTS AGE  nginx-65f88748fd-ckfhf 1/1 Running 0 43s  nginx-65f88748fd-hp99g 1/1 Running 0 43s  nginx-65f88748fd-wf62w 1/1 Running 0 5m42s    9）删除 deployment。  [root@master ~]# kubectl delete deployment nginx  deployment.extensions "nginx" deleted    10）删除 Service。  [root@master ~]# kubectl delete svc nginx  service "nginx" deleted    **【结论】（结果、分析）**  Kubernetes是开源的容器集群管理系统，可以实现容器集群的自动化部署、自动扩缩容、维护等功能。它既是一款容器编排工具，也是全新的基于容器技术的分布式架构领先方案。Kubernetes 在Docker技术的基础上，为容器化的应用提供部署运行、资源调度、服务发现和动态伸缩等功能，提高了大规模容器集群管理的便捷性。 |
| **三、指导教师评语及成绩：** |
| **评语：**  **成绩： 指导教师签名：徐峰**  **批阅日期：2023年 月 日** |