**搭建K8S环境**

**安装流程**

ssh登录，准备net.sh脚本

https://blog.csdn.net/sltin/article/details/93611808

https://github.com/easzlab/kubeasz/blob/master/docs/setup/00-planning\_and\_overall\_intro.md

|  |
| --- |
| Bash yum update -y && yum install python3 -y |

|  |
| --- |
| Bash yum install git -y  pip3 install pip --upgrade -i http://mirrors.aliyun.com/pypi/simple/ --trusted-host mirrors.aliyun.com pip3 install ansible==2.6.12 -i http://mirrors.aliyun.com/pypi/simple/ --trusted-host mirrors.aliyun.com |

//在.221执行，.221作为部署节点

|  |
| --- |
| Bash git clone -b 2.0.0 https://github.com/easzlab/kubeasz.git /etc/ansible   //进入ansible目录  cd /ect/ansible //创建hosts-file配置文件、将节点ip写入保存 vim hosts-file 10.176.62.221 10.176.62.222 10.176.62.223 10.176.62.224 10.176.62.225 ./tools/yc-ssh-key-copy.sh ./hosts-file 登录名 密码 |

|  |
| --- |
| Bash ansible all -m ping |

|  |
| --- |
| Bash yum install bridge-utils -y echo br\_netfilter > /etc/modules-load.d/br\_netfilter.conf modprobe br\_netfilter |

|  |
| --- |
| Bash export release=3.2.0 wget https://github.com/easzlab/kubeasz/releases/download/${release}/ezdown chmod +x ./ezdown |

|  |
| --- |
| Bash ./ezdown -D # 容器化运行kubeasz ./ezdown -S  # 创建新集群 k8s-01 docker exec -it kubeasz ezctl new k8s-01 |

集群配置文件在/etc/kubeasz/cluster/k8s-01/下，现在的做法是只修改hosts文件，如下

使用docker，flannel好像不用踩坑，之前用containerd有点恶心

|  |
| --- |
| Bash *# 'etcd' cluster should have odd member(s) (1,3,5,...)* [etcd] 10.176.62.221 10.176.62.223 10.176.62.222  *# master node(s)* [kube\_master] 10.176.62.221 10.176.62.222  *# work node(s)* [kube\_node] 10.176.62.225 10.176.62.222 10.176.62.223 10.176.62.224 10.176.62.221  *# [optional] harbor server, a private docker registry # 'NEW\_INSTALL': 'true' to install a harbor server; 'false' to integrate with existed one* [harbor] *#192.168.1.8 NEW\_INSTALL=false*  *# [optional] loadbalance for accessing k8s from outside* [ex\_lb] *#192.168.1.6 LB\_ROLE=backup EX\_APISERVER\_VIP=192.168.1.250 EX\_APISERVER\_PORT=8443 #192.168.1.7 LB\_ROLE=master EX\_APISERVER\_VIP=192.168.1.250 EX\_APISERVER\_PORT=8443*  *# [optional] ntp server for the cluster* [chrony] *#192.168.1.1*  [all:vars] *# --------- Main Variables --------------- # Secure port for apiservers* SECURE\_PORT="6443"  *# Cluster container-runtime supported: docker, containerd* CONTAINER\_RUNTIME="docker"  *# Network plugins supported: calico, flannel, kube-router, cilium, kube-ovn* CLUSTER\_NETWORK="flannel"  *# Service proxy mode of kube-proxy: 'iptables' or 'ipvs'* PROXY\_MODE="ipvs"  *# K8S Service CIDR, not overlap with node(host) networking* SERVICE\_CIDR="10.68.0.0/16"  *# Cluster CIDR (Pod CIDR), not overlap with node(host) networking* CLUSTER\_CIDR="172.20.0.0/16"  *# NodePort Range* NODE\_PORT\_RANGE="30000-32767"  *# Cluster DNS Domain* CLUSTER\_DNS\_DOMAIN="cluster.local"  *# -------- Additional Variables (don't change the default value right now) --- # Binaries Directory* bin\_dir="/opt/kube/bin"  *# Deploy Directory (kubeasz workspace)* base\_dir="/etc/kubeasz"  *# Directory for a specific cluster* cluster\_dir="{{ base\_dir }}/clusters/k8s-01"  *# CA and other components cert/key Directory* ca\_dir="/etc/kubernetes/ssl" |

|  |
| --- |
| Bash echo "alias dk='docker exec -it kubeasz'" >> /root/.bashrc source /root/.bashrc  # 一键安装，等价于执行docker exec -it kubeasz ezctl setup k8s-01 all dk ezctl setup k8s-01 01 dk ezctl setup k8s-01 02 dk ezctl setup k8s-01 03 dk ezctl setup k8s-01 04 dk ezctl setup k8s-01 05 dk ezctl setup k8s-01 06 dk ezctl setup k8s-01 07 |

**已经遇到的坑**

|  |
| --- |
| Bash # 查看kubelet状态以及日志信息 systemctl status kubelet journalctl -u kubelet # 发现报错 >> kubelet cgroup driver: "cgroupfs" is different from docker cgroup driver: "systemd" # 修改docker配置后重启，每个节点都要 vim /etc/docker/daemon.json >> "exec-opts": ["native.cgroupdriver=cgroupfs"], systemctl daemon-reload && systemctl restart docker |

时间不同步

|  |
| --- |
| Bash yum install chrony -y && systemctl enable chronyd reboot |

**Dashboard**

https://10.176.62.221:32518/

Token https://github.com/easzlab/kubeasz/blob/master/docs/guide/dashboard.md

|  |
| --- |
| Bash eyJhbGciOiJSUzI1NiIsImtpZCI6Il9Fam9zY2xEcTNldkVGTVRBNm1NNHRzZzVnMnZ4MUUydGRXcFlaYzhmNG8ifQ.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJrdWJlLXN5c3RlbSIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY291bnQvc2VjcmV0Lm5hbWUiOiJhZG1pbi11c2VyLXRva2VuLWd6NnN2Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bnQubmFtZSI6ImFkbWluLXVzZXIiLCJrdWJlcm5ldGVzLmlvL3NlcnZpY2VhY2NvdW50L3NlcnZpY2UtYWNjb3VudC51aWQiOiI4ZGJkNTI5Ny05N2Y3LTRkNDMtYWJjYy0wYTczYmJjNDczM2MiLCJzdWIiOiJzeXN0ZW06c2VydmljZWFjY291bnQ6a3ViZS1zeXN0ZW06YWRtaW4tdXNlciJ9.WBd2bmR8kASQQJBRsp5QZ7pP9k-1Hb-PggS7SZbwBKPmAhrWLb6wpA7Aw757pRCbDTArePqHJ5Jdiu2Y2kiqnZ\_Oot1yUxo5gOcLWoPoJ5IVKkgWKAqZzo-R8HYRqNumh8O\_BUjnh0vaLVx49Juql4YkRl7KK-q8WpExSqP-tAyFL0i9suGfHDWfhWC0Qp4MUkWa1Zm9pyR2G09aI\_Iuf8vMWHp8BIYcRlt74vnArytvx1Vtziz6P4HbKncUGFlkBg-yh6PAxEYTzBmAPl\_beUs7lOosCmjUWVi\_sRE0eP9gOz4jA\_KnINsuCmVwVnV\_pNmmJiV\_L77JVp\_5DUsJag |

**清理空间**

|  |
| --- |
| Bash du -h -x --max-depth=1 df -h  docker rm -v $(docker ps -a -q -f status=exited) docker rmi $(docker images -f "dangling=true" -q) docker system prune -a --volumes |

**扩容root**

|  |
| --- |
| Bash df -h tar cvf /tmp/home.tar /home umount /home lvremove /dev/mapper/centos-home #y lvextend -L +35G /dev/mapper/centos-root xfs\_growfs /dev/mapper/centos-root lvcreate -L 12G -n /dev/mapper/centos-home mkfs.xfs /dev/mapper/centos-home mount /dev/mapper/centos-home tar xvf /tmp/home.tar -C /home/ df -TH |

https://developer.aliyun.com/article/710634

**使用K8S PVC**

**所有节点**

|  |
| --- |
| Bash yum install -y nfs-utils rpcbind |

**master节点**

|  |
| --- |
| Bash mkdir /nfsdata1 mkdir /nfsdata2 mkdir /nfsdata3 chmod 777 /nfsdata1 chmod 777 /nfsdata2 chmod 777 /nfsdata3  vim /etc/exports /nfsdata1 \*(rw,no\_root\_squash,sync) /nfsdata2 \*(rw,no\_root\_squash,sync) /nfsdata3 \*(rw,no\_root\_squash,sync)  exportfs -rv  #手动加载 NFS 共享服务时，应该先启动 rpcbind，再启动 nfs systemctl start rpcbind && systemctl enable rpcbind systemctl start nfs && systemctl enable nfs  #查看 rpcbind 端口是否开启，rpcbind 服务默认使用 tcp 端口 111 netstat -anpt | grep rpcbind  showmount -e  # 测试用 echo '11111' > /nfsdata1/index.html echo '22222' > /nfsdata2/index.html echo '33333' > /nfsdata3/index.html  # 在client # sudo mount -t nfs <ip>:/<share dir> <target dir> # |

https://blog.csdn.net/shenyuanhaojie/article/details/122052862

https://blog.csdn.net/ShenHang\_/article/details/125116332

https://blog.csdn.net/arlen\_chen/article/details/115005015