**Kubernetes的安装及配置**

**官网地址：<https://kubernetes.io/zh-cn/releases/download>**

1. **安装前的准备工作**
2. **服务器资源硬件的准备**
   1. **硬件要求：**

* **内存：2GB以上**
* **CPU: 2核CPU以上**
* **硬盘: 20GB以上**

1. **服务器上环境资源的准备**
   1. **hosts配置(每台机器都需要设置),并设置hostname**

* **vim /etc/hosts**
* **hostnamectl set-hostname k8s-master01 #设置当前机器的hostname**
* **hostname #查看当前机器的hostname**
  1. **关闭防火墙**
* **systemctl stop firewalld**
* **systemctl disable firewalld**
  1. ****关闭**selinux**
* **setenforce 0**
* **sed -i "s/^SELINUX=enforcing/SELINUX=disabled/g" /etc/selinux/config**
* **reboot**
  1. **关闭系统的 Swap**
* **swapoff -a**
  1. ****设置路由,** **允许iptables检查桥接流量****
* **第一步：**
  + **cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf**
  + **overlay**
  + **br\_netfilter**
  + **EOF**
* **第二步：**
  + **sudo modprobe overlay**
  + **sudo modprobe br\_netfilter**
* **第三步：**
  + **cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf**
  + **net.bridge.bridge-nf-call-ip6tables = 1**
  + **net.bridge.bridge-nf-call-iptables = 1**
  + **net.ipv4.ip\_forward = 1**
  + **EOF**
* **第四步：**
  + **sudo sysctl -p /etc/sysctl.d/k8s.conf**
  + **sudo sysctl --system**
* **第五步骤：**
  + **cat /proc/sys/net/bridge/bridge-nf-call-iptables**
  + **cat /proc/sys/net/bridge/bridge-nf-call-ip6tables**
* **# 显示1则生效**
  1. **安装docker并修改配置**
* **sudo yum install docker-ce**
* **systemctl enable docker & systemctl start docker**
* **systemctl status docker**
* **vim /etc/docker/daemon.json #编辑docker的守护启动配置**
* **"exec-opts": ["native.cgroupdriver=systemd"] #编辑新增docker的驱动配置信息**

**【为什么要变更docker驱动？Kubernetes 推荐使用 systemd 来代替 cgroupfs因为systemd是Kubernetes自带的cgroup管理器, 负责为每个进程分配cgroups,但docker的cgroup driver默认是cgroupfs,这样就同时运行有两个cgroup控制管理器,当资源有压力的情况时,有可能出现不稳定的情况】**

* **根据选择安装CRI-O,cri-dockerd;因为docker安装附带安装了containerd; 这三个客户端都是 Kubernetes 中使用的容器运行时,这些容器运行时都是为了在 Kubernetes 集群中管理和运行容器而设计的**

1. **负载均衡器的准备**
2. **下载部署**
3. **适配Kubernetes相对应版本的kubeadm,kubectl,kubelet选择及安装**
   1. **插件详解**

* **kubelet是服务，调用container管理器，管理容器。**
* **kubectl是API，输入命令对k8s资源进行管理**
* **kubeadm是管理器，管理K8S的节点**
  1. **查看kubelet所有版本--#选择比较新的版本进行安装**

**yum list kubelet --showduplicates**

* 1. **安装 kubeadm、kubelet 和 kubectl，配置yum文件，因为国内无法直接访问google，这里需要将官网中的google的源改为国内源，以阿里云为例：**

**cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo**

**[kubernetes]**

**name=Kubernetes**

**baseurl=https://mirrors.aliyun.com/kubernetes/yum/repos/kubernetes-el7-aarch64/**

**enabled=1**

**gpgcheck=1**

**repo\_gpgcheck=1**

**gpgkey=https://mirrors.aliyun.com/kubernetes/yum/doc/yum-key.gpg https://mirrors.aliyun.com/kubernetes/yum/doc/rpm-package-key.gpg**

**EOF**

* 1. **执行安装，结果输出以下内容即为安装成功。**

**sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes**

* 1. **检查安装应用的信息**

**kubectl version & kubeadm version**

* 1. **检启动k8s服务**

**systemctl enable kubelet && systemctl start kubelet**

**systemctl status kubelet**

1. **初始化**
2. **初始化前的配置**
   1. **导出kubeadm默认配置文件**

**kubeadm config print init-defaults > kubeadm-init-config.yaml**

* 1. **编辑初始化文件**

1. **执行初始化**

**apiVersion: kubeadm.k8s.io/v1beta3**

**bootstrapTokens:**

**- groups:**

**- system:bootstrappers:kubeadm:default-node-token**

**token: abcdef.0123456789abcdef**

**ttl: 24h0m0s**

**usages:**

**- signing**

**- authentication**

**kind: InitConfiguration**

**localAPIEndpoint:**

**advertiseAddress: 172.31.47.134 #修改为第一台主机(master)的IP地址**

**bindPort: 6443**

**nodeRegistration:**

**criSocket: unix:///run/containerd/containerd.sock**

**imagePullPolicy: IfNotPresent**

**name: jollyDoy-Master01 #主机的名称**

**taints: null**

**---**

**apiServer:**

**timeoutForControlPlane: 4m0s**

**apiVersion: kubeadm.k8s.io/v1beta3**

**certificatesDir: /etc/kubernetes/pki**

**clusterName: kubernetes**

**controllerManager: {}**

**dns:**

**type: CoreDNS**

**etcd:**

**local:**

**dataDir: /var/lib/etcd**

**imageRepository: registry.cn-hangzhou.aliyuncs.com/google\_containers**

**kind: ClusterConfiguration**

**kubernetesVersion: 1.28.0 #安装的版本**

**#controlPlaneEndpoint: "172.31.47.134:6443" #多个主master的配置路径**

**networking:**

**dnsDomain: cluster.local**

**serviceSubnet: 10.96.0.0/12**

**podSubnet: 10.244.0.0/16 #分配的地址-配置成Calico网段**

**scheduler: {}**

**---**

**apiVersion: kubeproxy.config.k8s.io/v1alpha1**

**kind: KubeProxyConfiguration**

**mode: ipvs**

**---**

**apiVersion: kubelet.config.k8s.io/v1beta1**

**kind: KubeletConfiguration**

**cgroupDriver: systemd**

* **创建日志输出文件目录**

**mkdir logs**

* **初始化K8S并输出日志**

**kubeadm init --config=kubeadm-init-config.yaml --upload-certs | tee ./logs/kubeadm-init.log**

1. **查看安装信息**

**kubelet --version**

**kubeadm version**

**kubectl version**

1. **初始化之后的配置工作**

* **基础文件的创建及映射**
  + **说明:要开始使用集群，需要以普通用户身份运行以下命令:**

**mkdir -p $HOME/.kube**

**sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config**

**sudo chown $(id -u):$(id -g) $HOME/.kube/config**

* + **如果你是root用户，你可以运行:**

**export KUBECONFIG=/etc/kubernetes/admin.conf**

1. **网络搭建及配置(calico)**
2. **Calico的官网：<https://docs.tigera.io/calico/latest/getting-started/kubernetes/quickstart>**
3. **安装及配置**
   1. **下载calico.yaml文件**

**curl -L <https://docs.projectcalico.org/manifests/calico.yaml> -o calico.yaml**

* 1. **执行启动命令**

**kubectl apply -f calico.yaml**

* 1. **如果出现污点(taints),执行如下命令：**

**kubectl taint nodes --all node-role.kubernetes.io/control-plane-**

**kubectl taint nodes --all node-role.kubernetes.io/master-**

* 1. **检查所有pod和node的状态**

**kubectl get pods -A**

**kubectl get nodes -A**

1. **kubernetes-dashboard的安装及配置**
2. **下载kubernetes-dashboard.yaml 配置文件**

**curl https://raw.githubusercontent.com/kubernetes/dashboard/v2.7.0/aio/deploy/recommended.yaml -o recommended.yaml**

1. **编辑kubernetes-dashboard.yaml 配置文件**

* **打开kubernetes-dashboard.yaml文件：**

**vim kubernetes-dashboard.yaml**

* **根据以下内容编辑kubernetes-dashboard.yaml文件：**

**kind: Service**

**apiVersion: v1**

**metadata:**

**labels:**

**k8s-app: kubernetes-dashboard**

**name: kubernetes-dashboard**

**namespace: kubernetes-dashboard**

**spec:**

**type: NodePort # 新增**

**ports:**

**- port: 443**

**targetPort: 8443**

**nodePort: 30009 # 新增**

**selector:**

**k8s-app: kubernetes-dashboard**

1. **执行部署**

**kubectl apply -f recommended.yaml**

1. **创建访问账户，获取token**

* **创建一个admin-user.yaml的文件并编辑**

**# 新增命名空间**

**apiVersion: v1**

**kind: Namespace**

**metadata:**

**name: k8s-dashboard-auth**

**labels:**

**app: k8s-dashboard-auth**

**---**

**# 新增admin账户**

**apiVersion: v1**

**kind: ServiceAccount**

**metadata:**

**name: admin-user**

**namespace: k8s-dashboard-auth**

**labels:**

**app: k8s-dashboard-auth**

**---**

**# 新增创建角色并admin账户绑定管理员角色**

**apiVersion: rbac.authorization.k8s.io/v1**

**kind: ClusterRoleBinding**

**metadata:**

**name: admin-user**

**namespace: k8s-dashboard-auth**

**labels:**

**app: k8s-dashboard-auth**

**roleRef:**

**apiGroup: rbac.authorization.k8s.io**

**kind: ClusterRole**

**name: cluster-admin**

**subjects:**

**- kind: ServiceAccount**

**name: admin-user**

**namespace: k8s-dashboard-auth**

**---**

**#获取长寿命的不记名令牌，将令牌将保存在创建的secret中:**

**apiVersion: v1**

**kind: Secret**

**metadata:**

**name: admin-user-secret**

**namespace: k8s-dashboard-auth**

**labels:**

**app: k8s-dashboard-auth**

**annotations:**

**kubernetes.io/service-account.name: "admin-user"**

**type: kubernetes.io/service-account-token**

* **创建用户及其他信息**

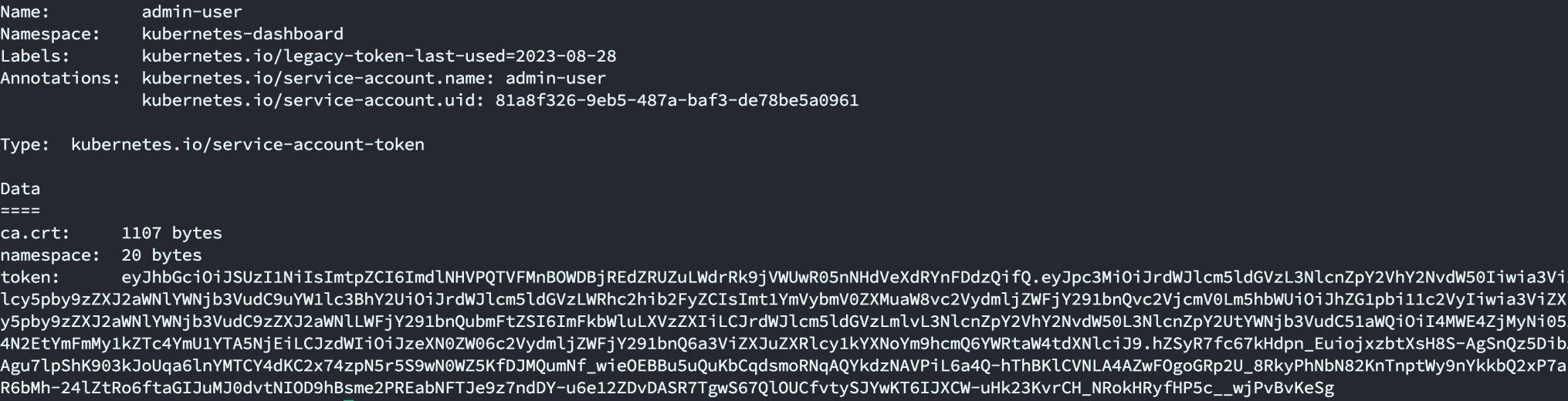
**kubectl apply -f dashboard-adminuser.yaml**

* **创建token;(这里的命名空间和用户名要和yaml文件保持一致)**

**kubectl -n k8s-dashboard-auth create token admin-user**

* **创建Secret后，我们可以执行以下命令来获取保存在Secret中的令牌**

**kubectl get secret admin-user -n k8s-dashboard-auth -o jsonpath={".data.token"} | base64 -d**



1. **metrics-server部署及应用**