

Kubernetes文章专栏地址: <http://blog.51cto.com/cloumn/detail/10>

## 官方提供Kubernetes部署3种方式

---

- **minikube**

Minikube是一个工具,可以在本地快速运行一个单点的Kubernetes,尝试Kubernetes或日常开发的用户使用。不能用于生产环境。

官方文档: <https://kubernetes.io/docs/setup/minikube/>

- **kubeadm**

kubeadm可帮助你快速部署一套kubernetes集群。kubeadm设计目的为新用户开始尝试kubernetes提供一种简单的方法。目前是Beta版。

官方文档: <https://kubernetes.io/docs/reference/setup-tools/kubeadm/kubeadm/> <https://kubernetes.io/docs/setup/independent/install-kubeadm/>

- **二进制包**

从官方下载发行版的二进制包,手动部署每个组件,组成Kubernetes集群。目前企业生产环境中主要使用该方式。下载地址: <https://github.com/kubernetes/kubernetes/blob/master/CHANGELOG-1.11.md#v1113>

## 1. 安装要求

---

- 操作系统
  - Ubuntu 16.04+
  - Debian 9
  - CentOS 7
  - RHEL 7
  - Fedora 25/26 (best-effort)
  - 其他
- 内存2GB+, 2核CPU+
- 集群节点之间可以通信
- 每个节点唯一主机名, MAC地址和product\_uuid
  - 检查MAC地址: 使用ip link或者ifconfig -a
  - 检查product\_uuid: cat /sys/class/dmi/id/product\_uuid
- 禁止swap分区。这样才能使kubelet正常工作

## 2. 准备环境

---

关闭防火墙:

```
# systemctl stop firewalld
# systemctl disable firewalld
```

关闭selinux:

```
# sed -i 's/enforcing/disabled/' /etc/selinux/config
# setenforce 0
```

关闭swap:

```
# swapoff -a # 临时
# vim /etc/fstab # 永久
```

添加主机名与IP对应关系:

```
# cat /etc/hosts
192.168.0.11 k8s-master
192.168.0.12 k8s-node1
192.168.0.13 k8s-node2
```

同步时间:

```
# yum install ntpdate -y
# ntpdate ntp.api.bz
```

### 3. 安装Docker

---

```
# yum install -y yum-utils device-mapper-persistent-data lvm2
```

```
# yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
```

目前docker最大支持docker-ce-17.03, 所以要指定该版本安装:

```
# yum install docker-ce-17.03.3.ce -y
```

如果提示container-selinux依赖问题, 先安装ce-17.03匹配版本:

```
# yum localinstall
https://download.docker.com/linux/centos/7/x86_64/stable/Packages/docker-ce-selinux-
17.03.3.ce-1.el7.noarch.rpm
```

```
# systemctl enable docker && systemctl start docker
```

### 4. 安装kubeadm, kubelet和kubectl

---

- kubeadm: 引导集群的命令
- kubelet: 集群中运行任务的代理程序
- kubectl: 命令行管理工具

#### 4.1 添加阿里云YUM软件源

```
# cat << EOF > /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://mirrors.aliyun.com/kubernetes/yum/repos/kubernetes-el7-x86_64
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
```

## 4.2 安装kubeadm, kubelet和kubectl

```
# yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes
# systemctl enable kubelet && systemctl start kubelet
```

注意：使用Docker时，kubeadm会自动检查kubelet的cgroup驱动程序，并/var/lib/kubelet/kubeadm-flags.env在运行时将其设置在文件中。如果使用的其他CRI，则必须在/etc/default/kubelet中cgroup-driver值修改为cgroupfs：

```
# cat /var/lib/kubelet/kubeadm-flags.env
KUBELET_KUBEADM_ARGS=--cgroup-driver=cgroupfs --cni-bin-dir=/opt/cni/bin --cni-conf-dir=/etc/cni/net.d --network-plugin=cni
# systemctl daemon-reload
# systemctl restart kubelet
```

## 5. 使用kubeadm创建单个Master集群

### 5.1 默认下载镜像地址在国外无法访问，先从准备好所需镜像

保存到脚本之间运行：

```
K8S_VERSION=v1.11.2
ETCD_VERSION=3.2.18
DASHBOARD_VERSION=v1.8.3
FLANNEL_VERSION=v0.10.0-amd64
DNS_VERSION=1.1.3
PAUSE_VERSION=3.1
# 基本组件
docker pull registry.cn-hangzhou.aliyuncs.com/google_containers/kube-apiserver-amd64:$K8S_VERSION
docker pull registry.cn-hangzhou.aliyuncs.com/google_containers/kube-controller-manager-amd64:$K8S_VERSION
docker pull registry.cn-hangzhou.aliyuncs.com/google_containers/kube-scheduler-amd64:$K8S_VERSION
docker pull registry.cn-hangzhou.aliyuncs.com/google_containers/kube-proxy-amd64:$K8S_VERSION
docker pull registry.cn-hangzhou.aliyuncs.com/google_containers/etcd-amd64:$ETCD_VERSION
```

```

docker pull registry.cn-hangzhou.aliyuncs.com/google_containers/pause:$PAUSE_VERSION
docker pull registry.cn-hangzhou.aliyuncs.com/google_containers/coredns:$DNS_VERSION
# 网络组件
docker pull quay.io/coreos/flannel:$FLANNEL_VERSION
# 修改tag
docker tag registry.cn-hangzhou.aliyuncs.com/google_containers/kube-apiserver-
amd64:$K8S_VERSION k8s.gcr.io/kube-apiserver-amd64:$K8S_VERSION
docker tag registry.cn-hangzhou.aliyuncs.com/google_containers/kube-controller-manager-
amd64:$K8S_VERSION k8s.gcr.io/kube-controller-manager-amd64:$K8S_VERSION
docker tag registry.cn-hangzhou.aliyuncs.com/google_containers/kube-scheduler-
amd64:$K8S_VERSION k8s.gcr.io/kube-scheduler-amd64:$K8S_VERSION
docker tag registry.cn-hangzhou.aliyuncs.com/google_containers/kube-proxy-
amd64:$K8S_VERSION k8s.gcr.io/kube-proxy-amd64:$K8S_VERSION
docker tag registry.cn-hangzhou.aliyuncs.com/google_containers/etcd-amd64:$ETCD_VERSION
k8s.gcr.io/etcd-amd64:$ETCD_VERSION
docker tag registry.cn-hangzhou.aliyuncs.com/google_containers/pause:$PAUSE_VERSION
k8s.gcr.io/pause:$PAUSE_VERSION
docker tag registry.cn-hangzhou.aliyuncs.com/google_containers/coredns:$DNS_VERSION
k8s.gcr.io/coredns:$DNS_VERSION

```

## 5.2 初始化Master

```
# kubeadm init --kubernetes-version=1.11.2 --pod-network-cidr=10.244.0.0/16 --apiserver-
advertise-address=192.168.0.11
```

...

Your Kubernetes master has initialized successfully!

To start using your cluster, you need to run (as a regular user):

```

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

```

You should now deploy a pod network to the cluster.

Run "kubectl apply -f [podnetwork].yaml" with one of the addon options listed at:

<http://kubernetes.io/docs/admin/addons/>

You can now join any number of machines by running the following on each node  
as root:

```
kubeadm join --token <token> <master-ip>:<master-port> --discovery-token-ca-cert-hash
sha256:<hash>
```

```

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

```

## 5.3 安装Pod网络 - 插件

```
# kubectl apply -f
https://raw.githubusercontent.com/coreos/flannel/v0.10.0/Documentation/kube-flannel.yml
```

## 5.4 加入工作节点

在Node节点切换到root账号：

格式：kubeadm join --token : --discovery-token-ca-cert-hash sha256:

```
# kubeadm join 192.168.0.11:6443 --token 6hk68y.0rdz1wdjyh85ntkr --discovery-token-ca-
cert-hash sha256:d1d3f59ae37fbd632707cbeb9b095d0d0b19af535078091993c4bc4d9d2a7782
```

## 6. kubernetes dashboard

```
# wget
https://raw.githubusercontent.com/kubernetes/dashboard/master/src/ deploy/recommended/kub
ernetes-dashboard.yml
```

修改镜像地址：

```
# registry.cn-hangzhou.aliyuncs.com/google_containers/kubernetes-dashboard-
amd64:v1.10.0
```

修改Service：

```
kind: Service
apiVersion: v1
metadata:
  labels:
    k8s-app: kubernetes-dashboard
  name: kubernetes-dashboard
  namespace: kube-system
spec:
  type: NodePort
  ports:
    - port: 443
      targetPort: 8443
      nodePort: 30001
  selector:
    k8s-app: kubernetes-dashboard
```

```
# kubectl apply -f kubernetes-dashboard.yml
```

创建一个管理员角色：

```
apiVersion: v1
kind: ServiceAccount
metadata:
```



## Kubernetes 仪表板

### ☐ Kubeconfig

请选择您已配置用来访问集群的 kubeconfig 文件，请浏览[配置对多个集群的访问](#)一节，了解更多关于如何配置和使用 kubeconfig 文件的信息

### ☒ 令牌

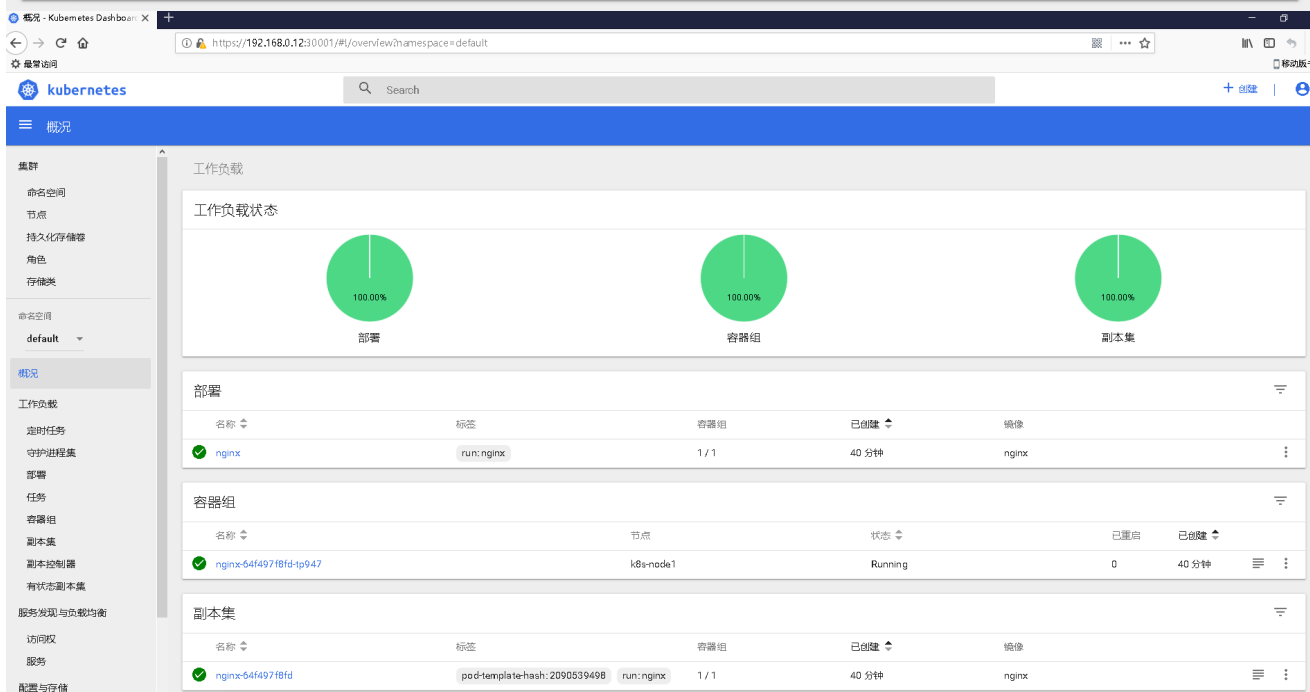
每个服务帐号都有一条保密字典保存持有者令牌，用来在仪表板登录，请浏览[验证](#)一节，了解更多关于如何配置和使用持有者令牌的信息

输入令牌

.....

登录

[跳过](#)



视频版: <https://ke.qq.com/course/266656>

### DevOps技术栈

专注于分享DevOps工具链  
及经验总结。

微信扫一扫



K8s技术交流群: 602965977