k8s 部署和使用

Name ： 曲中岭
Email:zlingqu@126.com
Q Q :441869115

第一章 部署准备

1.1 目的

使用 k8s 搭建 Docker 集群，实现相关功能，比如自动扩容、缩容、滚动更新等。

1.2 规划

OS ： CentOS_7.5 x64
Host1 ： 172.16.6.37(docker07),master 节点
Host2 ： 172.16.6.38(docker08),node 节点
Host3 ： 172.16.6.39(docker09),node 节点
Host4 ： 172.16.6.40(docker10),node 节点
Docker-ce ： 18.09.0

序号	类目	master 节点	node 节点	版本	安装方式
1	IP	172.16.6.37	172.16.6.38/39/40		
2	主机名	docker07	docker08/09/10		
3	docker	√	√	18.09.0	系统服务
4	kubeadm	√	√	v1.13.3	rpm
5	kubectl	√	√	v1.13.3	rpm
6	kubelet	√	√	v1.13.3	rpm
7	kube-proxy	√	√	v1.13.3	container
8	flannel	√	√	v1.13.3	container
9	pause	√	√	3.1	container
10	apiserver	√		v1.13.3	container
11	controller-manager	√		v1.13.3	container
12	scheduler	√		v1.13.3	container
13	etcd	√		3.2.24	container

pod 网络：10.244.0.0/16
service 网络：10.92.0.0/12
节点网络：172.20.0.0/16

1.3 k8s 的两种部署方式

方式 1

kubeadm 方式部署，k8s 可以把 k8s 自身的大部分应用管控起来，即运行于 pod 上，但是 kubelet 和 docker 不能这样实现自托管，这两个主机运行为守护进程，因此，只需要在所有主机都安装 kubelet 和 docker,构建 k8s 集群。相当于是自举。etcd 也是托管于 pod 上运行，使用 kubeadm 进行部署，安装过程相对简单。这些主件的 pod 一般为静态 pod (不属于 k8s 管理)，也可以运行为自托管的 pod.每个主机都要运行 flannel 这个主件，可以运行为 pod。flannel 为动态 pod。

kubeadm 的介绍可以查看如下链接

https://github.com/kubernetes/kubeadm/blob/master/docs/design/design_v1.10.md

安装步骤如下三步

1.master 和 node 安装 kubelet,kubeadm,docker

2.mster:kubeadm init，集群初始化

3.nodes:kubeadm join，node 节点加入集群

方式 2

手动配置，主节点和 node 都主要组件运行为系统级的守护进程，每一步都需要手动处理，如证书和配置过程都是用手动配置的。另外，这种方式在 github 上有 playbook 自动化实现

a).master:安装 apiserver,scheduler,controller-manager,etcd,flanel

b).node:安装 kublet,kub-proxy,docker(container engine),flannel,需要多个节点

c).etcd:安装 etcd 存储服务器，建议配置为高可用

这种方式，可以到 https://github.com/kubernetes/kubernetes/blob/master/CHANGELOG-1.11.md#downloads-for-v1112 下载相关的安装包，注意，master 或者 node 都是要安装 server 端的包。client 是交互时使用，也需要安装，不建议使用这种方式安装，难度较高

本文仅介绍使用 kubeadm 实现 k8s 集群安装

第二章 docker 安装

操作对象:

172.16.6.31
172.16.6.32
172.16.6.33

安装方法有很多，这里选择其中一种，rpm 方式。

2.1 安装

添加 docker 源:

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

或者使用国内阿里/清华的源:

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

yum-config-manager --add-repo https://mirrors.tuna.tsinghua.edu.cn/docker-ce/linux/centos/docker-ce.repo
```

从指定源安装 docker-ce ：

```
yum install docker-ce --enablerepo=docker-ce-stable -y

systemctl start docker
systemctl enable docker
```

查看是否开机运行：

```
systemctl list-unit-files|grep docker
```

2.2 确认

```
docker version
```

```
[root@docker07 ~]#
[root@docker07 ~]# docker version
Client:
 Version:           18.09.0
 API version:       1.39
 Go version:        go1.10.4
 Git commit:        4d60db4
 Built:             Wed Nov  7 00:48:22 2018
 OS/Arch:           linux/amd64
 Experimental:      false

Server: Docker Engine - Community
 Engine:
  Version:          18.09.0
  API version:      1.39 (minimum version 1.12)
  Go version:       go1.10.4
  Git commit:       4d60db4
  Built:            Wed Nov  7 00:19:08 2018
  OS/Arch:          linux/amd64
  Experimental:     false
[root@docker07 ~]#
```

2.3 ubuntu 安装（补充）

方法有很多，这里只说一种。

```
curl -sSL https://get.docker.com/ | sh
service start docker
sysv-rc-conf --list|grep docker
update-rc.d  docker  start 90 3 4 5 . stop 20 0 1 2 6 .
sysv-rc-conf --list|grep docker
docker version
```

第三章 kubeadm 等安装

操作主机:所有
所有主机安装 kubeadm、kubectl、kubelet

3.1 添加源

这里使用阿里云，也可使用其他源。另外，需要提醒的是，这几个包有个特别的地方，就是在下载后重新组装成的 rpm，而不是直接下载 rpm，所以必须在线安装。

```
cat >> /etc/yum.repos.d/k8s.repo << EOF
[k8s]
name=aliyun_k8s
baseurl=https://mirrors.aliyun.com/kubernetes/yum/repos/kubernetes-el7-x86_64/
enabled=1
gpgcheck=1
gpgkey=https://mirrors.aliyun.com/kubernetes/yum/doc/rpm-package-key.gpg
EOF
```

3.2 安装

```
yum install kubeadm
```

```
[root@docker09 ~]# yum install kubeadm
已加载插件：fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirrors.shu.edu.cn
 * extras: mirrors.aliyun.com
 * updates: mirrors.163.com
k8s
k8s/primary
k8s
正在解决依赖关系
--> 正在检查事务
---> 软件包 kubeadm.x86_64.0.1.13.3-0 将被 安装
--> 正在处理依赖关系 kubernetes-cni >= 0.6.0，它被软件包 kubeadm-1.13
--> 正在处理依赖关系 kubelet >= 1.6.0，它被软件包 kubeadm-1.13.3-0.x8
--> 正在处理依赖关系 kubectl >= 1.6.0，它被软件包 kubeadm-1.13.3-0.x8
--> 正在处理依赖关系 cri-tools >= 1.11.0，它被软件包 kubeadm-1.13.3-0
--> 正在检查事务
---> 软件包 cri-tools.x86_64.0.1.12.0-0 将被 安装
---> 软件包 kubectl.x86_64.0.1.13.3-0 将被 安装
---> 软件包 kubelet.x86_64.0.1.13.3-0 将被 安装
--> 正在处理依赖关系 socat，它被软件包 kubelet-1.13.3-0.x86_64 需要
```

自动安装依赖 kubectl 、kubelet、 kubernetes-cni

```
Package                    架构          版本                  源
==========================================================================
正在安装:
 kubeadm                   x86_64        1.13.3-0              k8s
为依赖而安装:
 conntrack-tools           x86_64        1.4.4-4.el7           base
 cri-tools                 x86_64        1.12.0-0              k8s
 kubectl                   x86_64        1.13.3-0              k8s
 kubelet                   x86_64        1.13.3-0              k8s
 kubernetes-cni            x86_64        0.6.0-0               k8s
 libnetfilter_cthelper     x86_64        1.0.0-9.el7           base
 libnetfilter_cttimeout    x86_64        1.0.0-6.el7           base
 libnetfilter_queue        x86_64        1.0.2-2.el7_2         base
 socat                     x86_64        1.7.3.2-2.el7         base

事务概要
==========================================================================
安装  1 软件包 (+9 依赖软件包)
```

第四章 部署集群

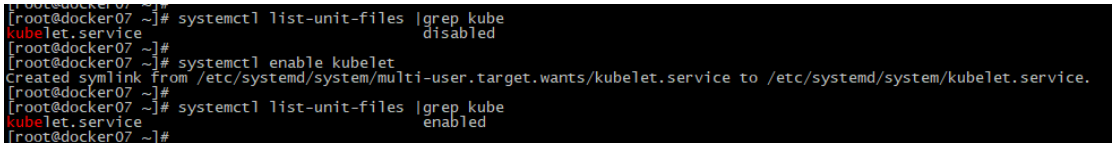
操作对象：

172.16.6.31
172.16.6.32
172.16.6.33

4.1 环境准备

4.1.1 kubelet 加入开机启动

```
systemctl enable kubelet
```



此时无法启动 kubelet，因为还未初始化完成，但需要将此服务加入开机启动

4.1.2 禁止 firewalld

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

4.1.3 调整内核参数

主要调整以下三个参数，并将其加入到/etc/rc.local 中。

```
echo 1 > /proc/sys/net/bridge/bridge-nf-call-iptables
echo 1 > /proc/sys/net/bridge/bridge-nf-call-ip6tables
echo 1 > /proc/sys/net/ipv4/ip_forward
```

4.1.4 host 配置

```
cat >> /etc/hosts << EOF
172.16.6.31 docker01
172.16.6.32 docker02
172.16.6.33 docker03
172.16.6.34 docker04
172.16.6.35 docker05
172.16.6.36 docker06
172.16.6.37 docker07
172.16.6.38 docker08
172.16.6.39 docker09
172.16.6.40 docker10
EOF
```

4.1.5 忽略 swap 错误

k8s 默认不支持 swap，如果开启了会 error 报错，处理方式有两种

方法 1：禁止 swap

```
swapoff -a && sed -i '/swap/d' /etc/fstab
```

方法 2：强制使用 swap

```
echo "KUBELET_EXTRA_ARGS=\"--fail-swap-on=false\"" > /etc/sysconfig/kubelet
```

并在初始化时添加如下参数

```
--ignore-preflight-errors=Swap
```

4.1.6 网络不通处理

初始化过程，默认会到 gcr.io/google_containers 站点拉取相关 k8s 的镜像信息，所需的镜像信息如 4.2.1 所列出。当前国内不能进行这些站点的访问，也就不能访问进行初始化安装。

解决方法 1：使用国外的代理服务器或则其他方法，使能够从该站点下载对应镜像

解决方法 2：使用 docker 官方的克隆镜像，方法如 4.4.2 所示。

本文档使用方法 2，方法 1 不再演示。

4.2 启动 master 节点

4.2.1 所需的镜像

```
[root@docker10 ~]# kubeadm config images list
k8s.gcr.io/kube-apiserver:v1.13.3
k8s.gcr.io/kube-controller-manager:v1.13.3
k8s.gcr.io/kube-scheduler:v1.13.3
k8s.gcr.io/kube-proxy:v1.13.3
k8s.gcr.io/pause:3.1
k8s.gcr.io/etcd:3.2.24
k8s.gcr.io/coredns:1.2.6
[root@docker10 ~]#
```

k8s.gcr.io/kube-apiserver:v1.13.3
k8s.gcr.io/kube-controller-manager:v1.13.3
k8s.gcr.io/kube-scheduler:v1.13.3
k8s.gcr.io/kube-proxy:v1.13.3
k8s.gcr.io/pause:3.1
k8s.gcr.io/etcd:3.2.24
k8s.gcr.io/coredns:1.2.6

注意 coredns、etcd 和 kube 模块的版本对应关系，可使用命令

```
kubeadm config images list
```

查到

4.2.2 拉取镜像

使用如下命令下载上述列出的镜像

```
docker pull mirrorgooglecontainers/kube-apiserver:v1.13.3
docker pull mirrorgooglecontainers/kube-controller-manager:v1.13.3
docker pull mirrorgooglecontainers/kube-scheduler:v1.13.3
docker pull mirrorgooglecontainers/kube-proxy:v1.13.3
docker pull mirrorgooglecontainers/pause:3.1
docker pull mirrorgooglecontainers/etcd:3.2.24
docker pull coredns/coredns:1.2.6
```

各模块还包含 64 位版本，比如 etcd 和 pause 可到如下页面查到。

https://hub.docker.com/r/mirrorgooglecontainers/etcd-amd64/tags

https://hub.docker.com/r/mirrorgooglecontainers/pause-amd64/tags

添加标签：

```
docker tag mirrorgooglecontainers/kube-apiserver:v1.13.3 k8s.gcr.io/kube-apiserver:v1.13.3
docker tag mirrorgooglecontainers/kube-controller-manager:v1.13.3 k8s.gcr.io/kube-controller-manager:v1.13.3
docker tag mirrorgooglecontainers/kube-scheduler:v1.13.3 k8s.gcr.io/kube-scheduler:v1.13.3
docker tag mirrorgooglecontainers/kube-proxy:v1.13.3 k8s.gcr.io/kube-proxy:v1.13.3
docker tag mirrorgooglecontainers/pause:3.1 k8s.gcr.io/pause:3.1
docker tag mirrorgooglecontainers/etcd:3.2.24 k8s.gcr.io/etcd:3.2.24
docker tag coredns/coredns:1.2.6 k8s.gcr.io/coredns:1.2.6
```

修改完成后，查看镜像

```
[root@docker07 ~]# docker images
REPOSITORY                                           TAG                 IMAGE ID            CREATED             SIZE
mirrorgooglecontainers/kube-apiserver                v1.13.3             fe242e556a99        2 weeks ago         181MB
k8s.gcr.io/kube-apiserver                            v1.13.3             fe242e556a99        2 weeks ago         181MB
mirrorgooglecontainers/kube-proxy                    v1.13.3             98db19758ad4        2 weeks ago         80.3MB
k8s.gcr.io/kube-proxy                                v1.13.3             98db19758ad4        2 weeks ago         80.3MB
mirrorgooglecontainers/kube-controller-manager       v1.13.3             0482f6400933        2 weeks ago         146MB
k8s.gcr.io/kube-controller-manager                   v1.13.3             0482f6400933        2 weeks ago         146MB
mirrorgooglecontainers/kube-scheduler                v1.13.3             3a6f709e97a0        2 weeks ago         79.6MB
k8s.gcr.io/kube-scheduler                            v1.13.3             3a6f709e97a0        2 weeks ago         79.6MB
coredns/coredns                                      1.3.1               eb516548c180        5 weeks ago         40.3MB
k8s.gcr.io/coredns                                   1.3.1               eb516548c180        5 weeks ago         40.3MB
mirrorgooglecontainers/etcd                          3.3.10              2c4adeb21b4f        2 months ago        258MB
k8s.gcr.io/etcd                                      3.3.10              2c4adeb21b4f        2 months ago        258MB
coredns/coredns                                      1.2.6               f59dcacceff4        3 months ago        40MB
k8s.gcr.io/coredns                                   1.2.6               f59dcacceff4        3 months ago        40MB
mirrorgooglecontainers/etcd                          3.2.24              3cab8e1b9802        5 months ago        220MB
k8s.gcr.io/etcd                                      3.2.24              3cab8e1b9802        5 months ago        220MB
k8s.gcr.io/pause                                     3.1                 da86e6ba6ca1        14 months ago       742kB
mirrorgooglecontainers/pause                         3.1                 da86e6ba6ca1        14 months ago       742kB
[root@docker07 ~]#
[root@docker07 ~]#
```

此时可以删除 mirrorgooglecontainers 相关的标签，我这里不再处理。

4.2.3 初始化集群

使用如下命令初始化集群

```
kubeadm init --kubernetes-version=v1.13.3 --pod-network-cidr=10.244.0.0/16 --service-cidr=10.96.0.0/12 --ignore-preflight-errors=Swap
```

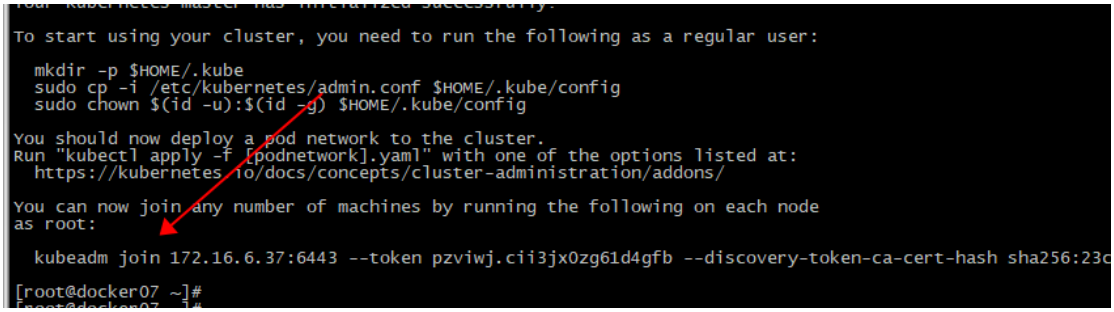
已经要进行 4.1 步骤，否则会有如下几个警告信息，

```
[root@docker07 ~]#
[root@docker07 ~]# kubeadm init --kubernetes-version=v1.13.3 --pod-network-cidr=10.244.0.0/16 --service-cidr=10.96.0.0/12
[init] Using Kubernetes version: v1.13.3
[preflight] Running pre-flight checks
        [WARNING Firewalld]: firewalld is active, please ensure ports [6443 10250] are open or your cluster may not function correctly
        [WARNING SystemVerification]: this Docker version is not on the list of validated versions: 18.09.0. Latest validated version: 18.06
        [WARNING Hostname]: hostname "docker07" could not be reached
        [WARNING Hostname]: hostname "docker07": lookup docker07 on 1.2.4.8:53: no such host
        [WARNING Service-Kubelet]: kubelet service is not enabled, please run 'systemctl enable kubelet.service'
error execution phase preflight: [preflight] Some fatal errors occurred:
        [ERROR Swap]: running with swap on is not supported. Please disable swap
[preflight] If you know what you are doing, you can make a check non-fatal with `--ignore-preflight-errors=...`
[root@docker07 ~]#
[root@docker07 ~]#
```

其中第二个警告信息说，kubeadm 目前支持最高版本是 18.06，而我们安装的是 18.09，这个警告忽略即可。

```
[root@docker07 ~]#
[root@docker07 ~]# kubeadm init --kubernetes-version=v1.13.3 --pod-network-cidr=10.244.0.0/16 --service-cidr=10.96.0.0/12 --ignore-preflight-errors=Swap
[init] Using Kubernetes version: v1.13.3
[preflight] Running pre-flight checks
        [WARNING Swap]: running with swap on is not supported. Please disable swap
        [WARNING SystemVerification]: this Docker version is not on the list of validated versions: 18.09.0. Latest validated version: 18.06
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Activating the kubelet service
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [docker07 kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.loc
.96.0.1 172.16.6.37]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
[certs] Generating "etcd/server" certificate and key
[certs] etcd/server serving cert is signed for DNS names [docker07 localhost] and IPs [172.16.6.37 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [docker07 localhost] and IPs [172.16.6.37 127.0.0.1 ::1]
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
[kubeconfig] Using kubeconfig folder "/etc/kubernetes"
[kubeconfig] Writing "admin.conf" kubeconfig file
[kubeconfig] Writing "kubelet.conf" kubeconfig file
```

初始化完成后，如下提示：



记录下上面这句话，用于 node 节点加入集群:

```
kubeadm join 172.16.6.37:6443 --token pzviwj.cii3jx0zg61d4gfb --discovery-token-ca-cert-hash sha256:23cbb3efbe8a2e2b73cc1442cf4b132b98511a451ffe14dacfe25b9594599c1a
```

4.3 启动 node 节点

4.3.1 安装必要包

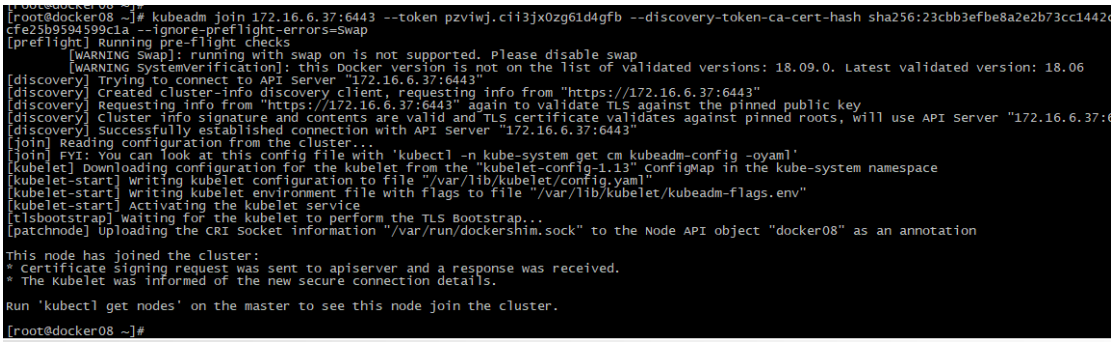
```
docker pull mirrorgooglecontainers/kube-proxy:v1.13.3
docker pull mirrorgooglecontainers/pause:3.1
docker tag mirrorgooglecontainers/kube-proxy:v1.13.3 k8s.gcr.io/kube-proxy:v1.13.3
docker tag mirrorgooglecontainers/pause:3.1 k8s.gcr.io/pause:3.1
```

4.3.2 加入集群

使用如下语句在 node 节点上执行即可加入集群，我这里所用了 swap

```
kubeadm join 172.16.6.37:6443 --token pzviwj.cii3jx0zg61d4gfb --discovery-token-ca-cert-hash sha256:23cbb3efbe8a2e2b73cc1442cf4b132b98511a451ffe14dacfe25b9594599c1a --ignore-preflight-errors=Swap
```

如下图：docker08 加入集群:



4.3.3 排错

如果出现如下错误

```
[root@docker09 ~]#
[root@docker09 ~]# kubeadm join 172.16.6.37:6443 --token pzviwj.cii3jx0zg61d4gfb --discovery-token-ca-cert-hash sh
cfe25b9594599c1a --ignore-preflight-errors=Swap
[preflight] Running pre-flight checks
        [WARNING Swap]: running with swap on is not supported. Please disable swap
        [WARNING SystemVerification]: this Docker version is not on the list of validated versions: 18.09.0. Lates
        [WARNING Service-Kubelet]: kubelet service is not enabled, please run 'systemctl enable kubelet.service'
[discovery] Trying to connect to API Server "172.16.6.37:6443"
[discovery] Created cluster-info discovery client, requesting info from "https://172.16.6.37:6443"
[discovery] Requesting info from "https://172.16.6.37:6443" again to validate TLS against the pinned public key
[discovery] Cluster info signature and contents are valid and TLS certificate validates against pinned roots, will
[discovery] Successfully established connection with API Server "172.16.6.37:6443"
[join] Reading configuration from the cluster...
[join] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -oyaml'
unable to fetch the kubeadm-config ConfigMap: failed to get config map: Unauthorized
[root@docker09 ~]#
[root@docker09 ~]#
```

unable to fetch the kubeadm-config ConfigMap: failed to get config map: Unauthorized

是因为 token 过期了，默认有效期 24 小时。

解决方法：在 master 节点上，使用如下命令重新生产新的 token

```
kubeadm token create
```

```
[root@docker07 ~]#
[root@docker07 ~]# kubeadm token create
e4l7o1.expvenkjvafg93yt
[root@docker07 ~]#
[root@docker07 ~]#
```

使用新的 token 重新加入集群，如下图

```
[root@docker09 ~]#
[root@docker09 ~]# kubeadm join 172.16.6.37:6443 --token e4l7o1.expvenkjvafg93yt --discovery-token-ca-cert-hash sha256:23cbb3efbe8a2e2b7
cfe25b9594599c1a --ignore-preflight-errors=Swap
[preflight] Running pre-flight checks
        [WARNING Swap]: running with swap on is not supported. Please disable swap
        [WARNING SystemVerification]: this Docker version is not on the list of validated versions: 18.09.0. Latest validated version: 1
        [WARNING Service-Kubelet]: kubelet service is not enabled, please run 'systemctl enable kubelet.service'
[discovery] Trying to connect to API Server "172.16.6.37:6443"
[discovery] Created cluster-info discovery client, requesting info from "https://172.16.6.37:6443"
[discovery] Requesting info from "https://172.16.6.37:6443" again to validate TLS against the pinned public key
[discovery] Cluster info signature and contents are valid and TLS certificate validates against pinned roots, will use API Server "172.1
[discovery] Successfully established connection with API Server "172.16.6.37:6443"
[join] Reading configuration from the cluster...
[join] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -oyaml'
[kubelet] Downloading configuration for the kubelet from the "kubelet-config-1.13" ConfigMap in the kube-system namespace
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Activating the kubelet service
[tlsbootstrap] Waiting for the kubelet to perform the TLS Bootstrap...
[patchnode] Uploading the CRI Socket information "/var/run/dockershim.sock" to the Node API object "docker09" as an annotation

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the master to see this node join the cluster.
```

4.4 配置网络

k8s 支持多种网络模型，比如

4.4.1 master 节点

使用如下语句安装：

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

4.4.2 node 节点

flannel 版本选择，查看如下：

https://quay.io/repository/coreos/flannel?tab=tags

使用如下命令下载镜像：

```
docker pull quay.io/coreos/flannel:v0.11.0-amd64
```

下载后，会被主节点调度，自动配置

4.5 部署 web-ui

到这里可以看到版本对应关系：

https://github.com/kubernetes/dashboard/releases

到这里选择镜像版本
https://hub.docker.com/r/mirrorgooglecontainers/kubernetes-dashboard-amd64/tags
比如我选择最新的 1.10.1 版本
docker pull mirrorgooglecontainers/kubernetes-dashboard-amd64:v1.10.1

4.6 观察

集群启动后，在 master 节点上观察集群运行状态是否和规划相符

4.5.1 配置环境变量

输入以下语句

```
echo "export KUBECONFIG=/etc/kubernetes/admin.conf" >> ~/.bash_profile
source ~/.bash_profile
```

若不进行这一步，执行任何 kubectl 命令都将出现以下错误

```
[root@docker07 ~]#
[root@docker07 ~]# kubectl get nodes
The connection to the server localhost:8080 was refused - did you specify the right host or port?
[root@docker07 ~]#
```

4.5.2 查看节点信息

```
[root@docker07 ~]# kubectl get nodes
NAME       STATUS     ROLES    AGE     VERSION
docker07   NotReady   master   2d23h   v1.13.3
docker08   NotReady   <none>   2d22h   v1.13.3
docker09   NotReady   <none>   80m     v1.13.3
docker10   NotReady   <none>   79m     v1.13.3
[root@docker07 ~]#
```

看到状态都是 NotReady 状态，因为未执行 4.4 步骤，执行后查看信息如下:

```
[root@docker07 ~]# kubectl get nodes
NAME       STATUS   ROLES    AGE     VERSION
docker07   Ready    master   3d1h    v1.13.3
docker08   Ready    <none>   3d1h    v1.13.3
docker09   Ready    <none>   3h23m   v1.13.3
docker10   Ready    <none>   3h22m   v1.13.3
[root@docker07 ~]#
```

第五章 测试

5.1 集群测试

待补充