

## 一、Pre-environmental preparation

### 1. Ready Server

Two Ubuntu servers are prepared here to ensure that the network connection of each server is normal. The specific environment configuration is as follows:

System	IP	Role	Memory	HostName
Ubuntu18.04	10.5.32.29	master	16G	ic3421-server-1
Ubuntu18.04	10.5.32.32	worker	8G	ic3421-server-2

### 2. docker related configuration

#### 2.1 Uninstall the old version(if exists)

```
$ apt-get remove docker docker-engine docker.io
```

#### 2.2 Update apt-get

```
$ add-apt-repository "deb [arch=amd64]  
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
```

```
$ apt-get update
```

#### 2.3 Install apt HTTPS support package and add GPG key

```
$ apt-get install \  
    apt-transport-https \  
    ca-certificates \  
    curl \  
    software-properties-common  
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | apt-key add -
```

#### 2.4 Install docker-ce

- Install latest version

```
$ apt-get install -y docker-ce
```

- Install specific version

```
 #(eg. 17.09.1~ce-0~ubuntu)
```

```
$ apt-get install -y docker-ce=17.09.1~ce-0~ubuntu
```

- Accept all forwarding ip packet

```
$ vim /lib/systemd/system/docker.service
```

```
#type in following code above 'ExecStart=xxx': (k8s network requirement)  
ExecStartPost=/sbin/iptables -I FORWARD -s 0.0.0.0/0 -j ACCEPT
```

- Start service and enable run on start-up

```
$ systemctl daemon-reload  
$ service docker restart
```

### 3. System Settings(All nodes)

#### 3.1 Shut down or disable firewall(allow free connection to servers)

```
$ ufw disable  
#check status  
$ ufw status
```

```
k8s@ic3421-server-1:~$ sudo ufw status  
Status: inactive
```

```
k8s@ic3421-server-2:~$ sudo ufw status  
Status: inactive
```

#### 3.2 Configure Host

```
#Configure host so that every node can resolve ip address by name  
$ vi /etc/hosts#add following code(replace ip address and server name with yours)
```

```
k8s@ic3421-server-1:~$ cat /etc/hosts  
127.0.0.1        localhost  
127.0.1.1        ic3421-server-1  
10.5.32.29       ic3421-server-1  
10.5.32.32       ic3421-server-2
```

```
k8s@ic3421-server-2:~$ cat /etc/hosts
127.0.0.1      localhost
127.0.1.1      ic3421-server-2
10.5.32.32     ic3421-server-2
10.5.32.29     ic3421-server-1
```

### 3.3 Shut Down Swap

```
swapoff -a
sudo sed -i 's/ swap / s/^\(.*\)$/#1/g' /etc/fstab
sed -i '0,/ExecStart=/s//Environment="KUBELET_EXTRA_ARGS=--cgroup-driver=cgroupfs"\n&/'
/etc/systemd/system/kubelet.service.d/10-kubeadm.conf
```

## 4. Kubernetes Node Configuration

### 4.1 Install kubelet, kubeadm and kubectl (All nodes)

#1. use domestic kubernetes mirror

① add apt-key:

```
curl -s https://mirrors.aliyun.com/kubernetes/apt/doc/apt-key.gpg | sudo apt-key
add -
```

--

② add ali mirror:

```
sudo vim /etc/apt/sources.list.d/kubernetes.list
```

```
deb https://mirrors.aliyun.com/kubernetes/apt/ kubernetes - xenial main
```

③ installation:

```
sudo apt update
```

```
sudo apt install -y kubelet kubeadm kubectl
```

```
sudo apt -mark hold kubelet kubeadm kubectl
```

#2. Use Google official kubernetes mirror

```
apt -get update && apt -get install -y apt - transport - https curl - s https://p
ackages.cloud.google.com/apt/doc/apt - key.gpg | apt - key add - cat
<<EOF>/etc/apt/sources.list.d/kubernetes.listdeb
http://apt.kubernetes.io/kubernetes - xenial mainEOF apt - get update apt - get
install -y kubelet kubeadm kubectl
```

## 4.2 Initialize Master

- Initialize Master node:

#Run as root

```
Kubeadm init --image-repository registry.aliyuncs.com/google_containers
--apiserver-advertise-address 10.5.32.29 --pod-network-cidr=10.244.0.0/16
```

**Note:**

`--apiserver-advertise-address`

Designate exactly which interface of Master for the communication with Cluster. Clear designation is suggested if Master has multiple interfaces. Otherwise kubeadm will use default interface.

`--pod-network-cidr`

Clarify the scope of Pod network. Kubernetes supports multiple network schemes with different rules. The cidr should be set as 10.244.0.0/16 for flannel network t. We will try other network in other practice.

For instance, a variable with default value of ‘k8s.gcr.io’ has been appended and the problem in downloading foreign mirrors has been solved in Canal 1.13. Set the mirror address to ‘registry.aliyuncs.com/google\_containers’ and recorded relative information as below.

### Master node initialization succeeded:

```
[control-plane] Creating static Pod manifest for "kube-scheduler"
[etcd] Creating static Pod manifest for local etcd in "/etc/kubernetes/manifests"
[wait-control-plane] Waiting for the kubelet to boot up the control plane as static Pods from directory "/etc/kubernetes/manifests". This can
s
[apiclient] All control plane components are healthy after 21.003061 seconds
[upload-config] storing the configuration used in ConfigMap "kubeadm-config" in the "kube-system" Namespace
[kubelet] Creating a ConfigMap "kubelet-config-1.14" in namespace kube-system with the configuration for the kubelets in the cluster
[upload-certs] Skipping phase. Please see --experimental-upload-certs
[mark-control-plane] Marking the node ic3421-server-1 as control-plane by adding the label "node-role.kubernetes.io/master="
[mark-control-plane] Marking the node ic3421-server-1 as control-plane by adding the taints [node-role.kubernetes.io/master:NoSchedule]
[bootstrap-token] Using token: 2yo099.p2bwdyogwobny9dd
[bootstrap-token] Configuring bootstrap tokens, cluster-info ConfigMap, RBAC Roles
[bootstrap-token] configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certificate credential
[bootstrap-token] configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstrap Token
[bootstrap-token] configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[bootstrap-token] creating the "cluster-info" ConfigMap in the "kube-public" namespace
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy
```

Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following 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 options listed at:  
<https://kubernetes.io/docs/concepts/cluster-administration/addons/>

Then you can join any number of worker nodes by running the following on each as root:

```
kubeadm join 10.5.32.29:6443 --token 2yo099.p2bwdyogwobny9dd \
--discovery-token-ca-cert-hash sha256:b04f6a1f73175b0ff29fcdc217228f08acfb92d50387b6a8b9fd26ef5db1d049
root@ic3421-server-1:~#
```

- Configure kubectl for common k8s user

```
root@ic3421-server-1:~# su k8s
k8s@ic3421-server-1:/root$ cd
k8s@ic3421-server-1:~$ cd
k8s@ic3421-server-1:~$ mkdir -p $HOME/.kube
k8s@ic3421-server-1:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
[sudo] password for k8s:
cp: overwrite '/home/k8s/.kube/config'? y
k8s@ic3421-server-1:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
k8s@ic3421-server-1:~$
```

- Use flannel

Kubernetes was deployed on Master node through kubeadm in the above paragraph. But in order to get kubernetes Cluster to work, pod network has to be installed. Otherwise connection will not be reached in pod. Though kubernetes supports multiple network schemes, the following commands have to be executed to use flannel

```
kubectl apply -f [podnetwork].yaml
```

Yaml netlist:

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

yaml with flannel deployed:

<https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml>

```
k8s@ic3421-server-1:/etc/kubernetes$ kubectl apply -f kube-flannel.yml
podsecuritypolicy.extensions/psp.flannel.unprivileged created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.extensions/kube-flannel-ds-amd64 created
daemonset.extensions/kube-flannel-ds-arm64 created
daemonset.extensions/kube-flannel-ds-arm created
daemonset.extensions/kube-flannel-ds-ppc64le created
daemonset.extensions/kube-flannel-ds-s390x created
k8s@ic3421-server-1:/etc/kubernetes$
```

## 4.3 Register and add Work node

Run the following command on node 'ic3421-server-2' to register it on Cluster:

#Run as root

swapoff -a #disable swap

```
Kubeadm join 10.5.32.29:6443 --token 2yo099.p2bwdyogwobny9dd
--discovery-token-ca-cert-hash
sha256:b04f6a1f73175b0ff29fcdc217228f08acfb92d50387b6a8b9fd26ef5db1d049
```

Feedback for the this procedure:

```
root@ic3421-server-2:/home/k8s# kubectl join 10.5.32.29:6443 --token 2yo099.p2bwdyogwobny9dd --discovery-token-ca-cert-hash
sha256:b04f6a1f73175b0ff29fcdc217228f08acfb92d50387b6a8b9fd26ef5db1d049
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -oyaml'
[kubelet-start] Downloading configuration for the kubelet from the "kubelet-config-1.14" 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
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...

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

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.
```

## 4.5 Check Cluster status

Switch to Master node and run kubectl command to check Cluster status. Kubernetes Cluster has been successfully deployed if you get the message below:

#kubectl get node

```
k8s@ic3421-server-1:~$ kubectl get node
NAME                STATUS    ROLES    AGE    VERSION
ic3421-server-1     Ready    master   16h    v1.14.2
ic3421-server-2     Ready    <none>   16h    v1.14.3
```