

kubernetes 单机安装

作者: [boskt](#)

原文链接: <https://ld246.com/article/1539827012401>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

服务器规划

角色	IP	环境
MASTER apiserver, kube-controller-manager, kube-scheduler, etcd	192.168.112.110	kube
NODE , kube-proxy, docker, flannel	192.168.112.110	kubele

安装etcd

1.检测防火墙

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

2.安装etcd服务

```
# mkdir -p /var/lib/etcd && mv etcd/{etcd,etcdctl} /usr/bin/  
# vi /usr/lib/systemd/system/etcd.service
```

```
[Unit]  
Description=Etcd Server  
After=network.target
```

```
[Service]  
Type=simple  
WorkingDirectory=/var/lib/etcd/  
EnvironmentFile=-/etc/etcd/etcd.conf  
ExecStart=/usr/bin/etcd  
Type=notify
```

```
[Install]  
WantedBy=multi-user.target
```

```
# mkdir -p /etc/etcd/ && vi /etc/etcd/etcd.conf
```

```
# [member]  
ETCD_NAME=etcd1  
ETCD_DATA_DIR="/var/lib/etcd"  
ETCD_LISTEN_CLIENT_URLS="http://192.168.112.110:2379,http://127.0.0.1:2379"  
ETCD_ADVERTISE_CLIENT_URLS="http://192.168.112.110:2379,http://127.0.0.1:2379"
```

```
#[cluster]  
ETCD_LISTEN_PEER_URLS="http://192.168.112.110:2380"  
ETCD_INITIAL_ADVERTISE_PEER_URLS="http://192.168.112.110:2380"  
ETCD_INITIAL_CLUSTER="etcd1=http://192.168.112.110:2380"  
ETCD_INITIAL_CLUSTER_STATE="new"  
ETCD_INITIAL_CLUSTER_TOKEN="etcd-cluster"
```

```
# systemctl daemon-reload  
# systemctl enable etcd.service
```

```
# systemctl restart etcd
# etcdctl cluster-health
```

安装Flannel

```
etcdctl --endpoints="http://192.168.112.110:2379" set \
/coreos.com/network/config '{ "Network": "10.1.0.0/16", "Backend": {"Type": "vxlan"}}'

# unzip flannel.zip && cd flannel
# mkdir -p /opt/kubernetes/bin && mv flanneld mk-docker-opts.sh /opt/kubernetes/bin

# mkdir -p /opt/kubernetes/cfg/ && vi /opt/kubernetes/cfg/flanneld

FLANNEL_OPTIONS="--etcd-endpoints=http://192.168.112.110:2379"
```

配置flanneld

```
# vi /usr/lib/systemd/system/flanneld.service
```

```
[Unit]
```

```
Description=Flanneld overlay address etcd agent
After=network-online.target network.target
Before=docker.service
```

```
[Service]
```

```
Type=notify
EnvironmentFile=/opt/kubernetes/cfg/flanneld
ExecStart=/opt/kubernetes/bin/flanneld --ip-masq $FLANNEL_OPTIONS
ExecStartPost=/opt/kubernetes/bin/mk-docker-opts.sh -k DOCKER_NETWORK_OPTIONS -d /
un/flannel/subnet.env
Restart=on-failure
```

```
[Install]
```

```
WantedBy=multi-user.target
```

```
# mv /usr/lib/systemd/system/docker.service /usr/lib/systemd/system/docker.service.bak
```

```
# vi /usr/lib/systemd/system/docker.service
```

```
[Unit]
```

```
Description=Docker Application Container Engine
Documentation=https://docs.docker.com
After=network-online.target firewalld.service
Wants=network-online.target
```

```
[Service]
```

```
Type=notify
EnvironmentFile=/run/flannel/subnet.env
ExecStart=/usr/bin/dockerd $DOCKER_NETWORK_OPTIONS
ExecReload=/bin/kill -s HUP $MAINPID
LimitNOFILE=infinity
LimitNPROC=infinity
LimitCORE=infinity
TimeoutStartSec=0
```

```
Delegate=yes  
KillMode=process  
Restart=on-failure  
StartLimitBurst=3  
StartLimitInterval=60s
```

```
[Install]  
WantedBy=multi-user.target
```

```
# systemctl daemon-reload  
# systemctl start flanneld  
# systemctl enable flanneld  
# systemctl restart docker
```

检查是否生效:

```
# ps -ef |grep docker  
# ip addr  
# etcdctl ls /coreos.com/network/subnets  
# etcdctl get /coreos.com/network/subnets/10.1.22.0-24
```

安装MASTER节点

配置kube-apiserver

```
# cp kubernetes/server/bin/{kube-apiserver,kube-scheduler,kubectl,kube-controller-manager}  
/usr/bin/
```

```
# vi /usr/lib/systemd/system/kube-apiserver.service
```

```
[Unit]  
Description=Kubernetes API Server  
Documentation=https://github.com/kubernetes/kubernetes  
After=etcd.service  
Wants=etcd.service
```

```
[Service]  
EnvironmentFile=/etc/kubernetes/apiserver  
ExecStart=/usr/bin/kube-apiserver $KUBE_API_ARGS  
Restart=on-failure  
Type=notify  
LimitNOFILE=65536
```

```
[Install]  
WantedBy=multi-user.target
```

```
# mkdir -p /etc/kubernetes/ && vi /etc/kubernetes/apiserver
```

```
KUBE_API_ARGS="--v=2 \  
--logtostderr=false \  
--log-dir=/var/log/kubernetes \  
--etcd-servers=http://127.0.0.1:2379 \  
--insecure-bind-address=0.0.0.0 \  
--insecure-port=8080 \  

```

```
--service-cluster-ip-range=169.169.0.0/16 \  
--service-node-port-range=1-65535 \  
--admission_control=\  
NamespaceLifecycle,LimitRanger,SecurityContextDeny,ServiceAccount,ResourceQuota"
```

```
# ServiceAccount
```

```
# systemctl daemon-reload  
# systemctl enable kube-apiserver  
# systemctl restart kube-apiserver  
# curl http://192.168.112.110:8080
```

配置kube-controller

```
# vi /usr/lib/systemd/system/kube-controller-manager.service
```

```
[Unit]
```

```
Description=Kubernetes Controller Manager  
Documentation=https://github.com/kubernetes/kubernetes  
After=kube-apiserver.service  
Requires=kube-apiserver.service
```

```
[Service]
```

```
EnvironmentFile=/etc/kubernetes/kube-controller-manager  
ExecStart=/usr/bin/kube-controller-manager $KUBE_CONTROLLER_MANAGER_OPTS  
Restart=on-failure  
LimitNOFILE=65536
```

```
[Install]
```

```
WantedBy=multi-user.target
```

```
# vi /etc/kubernetes/kube-controller-manager
```

```
KUBE_CONTROLLER_MANAGER_OPTS="--v=2 \  
--log-dir=/var/log/kubernetes \  
--logtostderr=false \  
--service_account_private_key_file=/var/run/kubernetes/apiserver.key \  
--master=http://127.0.0.1:8080"
```

```
# systemctl daemon-reload  
# systemctl enable kube-controller-manager  
# systemctl restart kube-controller-manager
```

配置kube-scheduler

```
# vi /usr/lib/systemd/system/kube-scheduler.service
```

```
[Unit]
```

```
Description=Kubernetes Scheduler  
Documentation=https://github.com/kubernetes/kubernetes  
After=kube-apiserver.service  
Requires=kube-apiserver.service
```

```
[Service]
```

```
EnvironmentFile=/etc/kubernetes/scheduler
ExecStart=/usr/bin/kube-scheduler $KUBE_SCHEDULER_OPTS
Restart=on-failure
LimitNOFILE=65536
```

```
[Install]
WantedBy=multi-user.target
```

```
# vi /etc/kubernetes/scheduler
```

```
KUBE_SCHEDULER_OPTS="--v=2 \
--log-dir=/var/log/kubernetes \
--logtostderr=false \
--master=http://127.0.0.1:8080"
```

```
# systemctl daemon-reload
# systemctl enable kube-scheduler
# systemctl restart kube-scheduler
```

验证

```
# systemctl status kube-apiserver
# systemctl status kube-scheduler
# systemctl status kube-controller-manager
# curl http://192.168.112.110:8080
```

安装NODE节点

```
# mkdir -p /var/lib/kubelet
# cp kubernetes/server/bin/{kubelet,kube-proxy} /usr/bin/
```

配置kubelet

```
# vi /usr/lib/systemd/system/kubelet.service
```

```
[Unit]
Description=Kubernetes Kubelet
After=docker.service
Requires=docker.service
```

```
[Service]
WorkingDirectory=/var/lib/kubelet
EnvironmentFile=/etc/kubernetes/kubelet
ExecStart=/usr/bin/kubelet $KUBELET_ARGS
Restart=on-failure
```

```
[Install]
WantedBy=multi-user.target
```

```
# vi /etc/kubernetes/kubelet
```

```
KUBELET_ARGS="--v=2 \
--api-servers=http://192.168.112.110:8080 \
--hostname-override=192.168.112.110 \
```

```
--logtostderr=false \  
--log-dir=/var/log/kubernetes"
```

```
# systemctl daemon-reload  
# systemctl enable kubelet  
# systemctl restart kubelet
```

配置kube-proxy

```
# vi /usr/lib/systemd/system/kube-proxy.service
```

```
[Unit]
```

```
Description=Kubernetes Proxy  
After=network.target  
Requires=network.target
```

```
[Service]
```

```
EnvironmentFile=/etc/kubernetes/proxy  
ExecStart=/usr/bin/kube-proxy $KUBE_PROXY_ARGS  
Restart=on-failure  
LimitNOFILE=65536
```

```
[Install]
```

```
WantedBy=multi-user.target
```

```
# vi /etc/kubernetes/proxy
```

```
KUBE_PROXY_ARGS="--v=2 \  
--log-dir=/var/log/kubernetes \  
--logtostderr=false \  
--master=http://192.168.112.110:8080"
```

```
# systemctl daemon-reload  
# systemctl enable kube-proxy  
# systemctl restart kube-proxy
```

验证

```
# systemctl status kubelet  
# systemctl status kube-proxy  
# kubectl get nodes
```