

步骤：

准备三台 linux 虚拟机，安装完成后设置 ip。或者使用我共享的虚拟机，vmware，复制成 3 台，分别设置好虚拟机。使用 NAT 方式。

虚拟机软件：windows 使用 vmware workstation 9，网盘里有共享的安装文件和注册机  
mac 使用 vmware fushion，可以搜索下载

镜像文件：可以使用我共享的 centos7 虚拟机 ovf 文件导入，需要重新设置 ip，copy 一下成为 N 台虚拟机即可。安装完后记得做个快照，防止将系统搞坏。也可以使用我提供的 centos7 的安装镜像自己重新安装。

下载地址：

链 接： <https://pan.baidu.com/s/1O8SxwHolh3Kb6nQplVnu-g> 密 码： 5u8j

## centos7 虚拟机的配置：

centos7 虚拟机的密码：root/123qwe

### 1.1 网络模式设置【vmware fushion mac 版】：

#### vmware fusion 的网络 NAT 模式[MAC OS]

进去以下目录：

`cd /Library/Preferences/VMware Fusion`

```
[VMware Fusion] $ ll
total 40
drwxr-xr-x 10 root wheel 320B 7 30 21:28 .
drwxr-xr-x 60 root wheel 1.9K 7 30 22:08 ..
-r--r--r-- 1 root wheel 31B 7 30 21:28 lastLocationUsed
-rw-r--r-- 1 root wheel 548B 7 30 2019 license-fusion-100-e3-201704
-rw-r--r-- 1 root wheel 487B 12 15 2017 networking
-rw-r--r-- 1 root wheel 463B 7 30 2019 networking.bak
-rw-r--r-- 1 root wheel 487B 7 14 10:09 networking.bak.0
drwxr-xr-x@ 10 root wheel 320B 7 30 21:28 thnucInt
drwxr-xr-x 4 root wheel 128B 7 30 2019 vmnet1
drwxr-xr-x 7 root wheel 224B 7 14 10:10 vmnet8
```

将 vmnet8 的配置改为如下,编辑 networking 如果没有 vmnet8 则新增一个 : 按照以下配置 :

```
answer VNET_1_DHCP yes
answer VNET_1_DHCP_CFG_HASH 9F5550209301981B6E02A89215830CC511C9169
answer VNET_1_HOSTONLY_NETMASK 255.255.255.0
answer VNET_1_HOSTONLY_SUBNET 192.168.177.0
answer VNET_1_VIRTUAL_ADAPTER yes
answer VNET_8_DHCP yes
answer VNET_8_DHCP_CFG_HASH 1480098C3D332805183F1FAD89EA06E3D
answer VNET_8_HOSTONLY_NETMASK 255.255.255.0
answer VNET_8_HOSTONLY_SUBNET 192.168.8.0
answer VNET_8_NAT yes
answer VNET_8_VIRTUAL_ADAPTER yes
add_bridge_mapping en0 2
```

回到当前目录, 进入 vmnet8 子目录 :

```
[VMware Fusion] $ pwd
/Library/Preferences/VMware Fusion
[VMware Fusion] $ ll
total 40
drwxr-xr-x 10 root wheel 320B 7 30 21:28 .
drwxr-xr-x 60 root wheel 1.9K 7 30 22:13 ..
-r--r--r-- 1 root wheel 31B 7 30 21:28 lastLocationUse
-rw-r--r-- 1 root wheel 548B 7 30 2019 license-fusion-10
-rw-r--r-- 1 root wheel 487B 12 15 2017 networking
-rw-r--r-- 1 root wheel 463B 7 30 2019 networking.bak
-rw-r--r-- 1 root wheel 487B 7 14 10:09 networking.bak.0
drwxr-xr-x@ 10 root wheel 320B 7 30 21:28 thnucInt
drwxr-xr-x 4 root wheel 128B 7 30 2019 vmnet1
drwxr-xr-x 7 root wheel 224B 7 14 10:10 vmnet8
[VMware Fusion] $
```

该目录下有以下文件 :

```
[VMware Fusion] $ cd vmnet8/
[vmnet8] $ ll
total 40
drwxr-xr-x  7 root  wheel  224B  7 14 10:10 .
drwxr-xr-x 10 root  wheel  320B  7 30 21:28 ..
-rw-r--r--  1 root  wheel  1.6K  7 14 10:17 dhcpd.conf
-rw-r--r--  1 root  wheel  1.6K  7 14 10:17 dhcpd.conf.bak
-rw-r--r--  1 root  wheel  1.5K  7 14 10:17 nat.conf
-rw-r--r--  1 root  wheel  1.5K  7 14 10:17 nat.conf.bak
-rw-r--r--  1 root  wheel   18B  7 30 21:28 nat.mac
[vmnet8] $
```

首先，修改 dhcpd.conf 文件内容如下：

```
# Written at: 07/14/2020 10:17:51
allow unknown-clients;
default-lease-time 1800;          # default is 30 minutes
max-lease-time 7200;             # default is 2 hours

subnet 192.168.8.0 netmask 255.255.255.0 {
    range 192.168.8.128 192.168.8.254;
    option broadcast-address 192.168.8.255;
    option domain-name-servers 192.168.8.2;
    option domain-name localdomain;
    default-lease-time 1800;      # default is 30 minutes
    max-lease-time 7200;         # default is 2 hours
    option netbios-name-servers 192.168.8.2;
    option routers 192.168.8.2;
}
host vmnet8 {
    hardware ethernet 00:50:56:C0:00:08;
    fixed-address 192.168.8.1;
    option domain-name-servers 0.0.0.0;
    option domain-name "";
    option routers 0.0.0.0;
}
```

修改 nat.conf 如下：

```
# VMware NAT configuration file
# Manual editing of this file is not recommended. Using UI is preferred.

[host]

# NAT gateway address
ip = 192.168.8.2
```

```
netmask = 255.255.255.0

# VMnet device if not specified on command line
device = vmnet8

# Allow PORT/EPRT FTP commands (they need incoming TCP stream ...)
activeFTP = 1

# Allows the source to have any OUI. Turn this on if you change the OUI
# in the MAC address of your virtual machines.
allowAnyOUI = 1

# Controls if (TCP) connections should be reset when the adapter they are
# bound to goes down
resetConnectionOnLinkDown = 1

# Controls if (TCP) connection should be reset when guest packet's destination
# is NAT's IP address
resetConnectionOnDestLocalHost = 1

# Controls if enable nat ipv6
natIcmp6Enable = 0

# Controls if enable nat ipv6
natIcmp6Prefix = fd15:4ba5:5a2b:1008::/64

[tcp]

# Value of timeout in TCP TIME_WAIT state, in seconds
timeWaitTimeout = 30

[udp]

# Timeout in seconds. Dynamically-created UDP mappings will be purged if
# idle for this duration of time 0 = no timeout, default = 60; real
# value might be up to 100% longer
timeout = 60

[netbios]

# Timeout for NBNS queries.
nbnsTimeout = 2

# Number of retries for each NBNS query.
nbnsRetries = 3
```

```
# Timeout for NBDS queries.
nbnsTimeout = 2

# Number of retries for each NBNS query.
nbnsRetries = 3

# Timeout for NBDS queries.
nbdsTimeout = 3

[incomingtcp]

# Use these with care - anyone can enter into your VM through these...
# The format and example are as follows:
#<external port number> = <VM's IP address>:<VM's port number>
#8080 = 172.16.3.128:80

[incomingudp]

# UDP port forwarding example
#6000 = 172.16.3.0:6001
```

## 然后重启虚拟网络：

```
sudo /Applications/VMware\ Fusion.app/Contents/Library/vmnet-cli --stop
sudo /Applications/VMware\ Fusion.app/Contents/Library/vmnet-cli --start
```

## 查看一下 ifconfig：

```
media: autoselect (100baseTX <full duplex>)
status: active
vmnet1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
ether 00:50:56:c0:00:01
inet 192.168.177.1 netmask 0xffffffff broadcast 192.168.177.255
vmnet8: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
ether 00:50:56:c0:00:08
inet 192.168.8.1 netmask 0xffffffff broadcast 192.168.8.255
[vmnet8] $
```

查看 vmnet8 是否有 ip 且为 192.168.8.1，且能 ping 通说明配置成功：

```
[vmnet8] $ ping 192.168.8.1
PING 192.168.8.1 (192.168.8.1): 56 data bytes
64 bytes from 192.168.8.1: icmp_seq=0 ttl=64 time=0.050 ms
64 bytes from 192.168.8.1: icmp_seq=1 ttl=64 time=0.049 ms
64 bytes from 192.168.8.1: icmp_seq=2 ttl=64 time=0.054 ms
^C
192.168.8.1 ping statistics:
```

然后在虚拟机菜单中将网络设置为 NAT 网络即可：



## 2 检查 k8s 是否正常启动

### 2.1 检查方式

#### 确保 apiserver 启动：

apiserver 作为总线，所有的 pod 和 kubelet 都要和其打交道，需要确保 apiserver 启动：

```
[root@vm81 ~]# ps -efl | grep apiserver
root      10462  10408  3 14:43 ?                00:00:43 kube-apiserver --advertise-address=192.168.1.100 --enable-admission-plugins=NodeRestriction --enable-bootstrap-token-auth=true --etcd-client-cert=/etc/kubernetes/pki/apiserver-etcd-client.crt --etcd-client-key=/etc/kubernetes/pki/apiserver-etcd-client.key --etcd-endpoints=https://192.168.1.100:2379 --etcd-servers=https://192.168.1.100:2379 --kubelet-client-cert=/etc/kubernetes/pki/apiserver-kubelet-client.crt --kubelet-client-key=/etc/kubernetes/pki/apiserver-kubelet-client.key --kubelet-qos-protection-policy=BestEffort --kubelet-server-cert=/etc/kubernetes/pki/apiserver-kubelet-server.crt --kubelet-server-key=/etc/kubernetes/pki/apiserver-kubelet-server.key --requestheader-extra-headers-prefix=X-Remote-User --secure-port=6443 --service-account-key-file=/etc/kubernetes/pki/sa.pub --service-account-signing-key-file=/etc/kubernetes/pki/apiserver.key
root      31520  12159  0 15:02 pts/0        00:00:00 grep --color=auto apiserver
```

apiserver 是部署在 k8s pod 中的，如果 apiserver 没有启动，原因可能有：

1. kubelet 没有启动
2. kubelet 启动了，但是 pod 没有启动

#### 确保 kubelet 启动：

```
[root@vm81 ~]# ps -efl | grep kubelet
root      8553    1  2 14:43 ?                00:00:34 /usr/bin/kubelet --bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kubeconfig.yaml --cgroup-driver=systemd --network-plugin=cni --pod-infra-container-image=registry.k8s.io/pause:3.1
```

如果没有启动：执行

**systemctl restart kubelet**

确保 k8s 的 node 处于 ready 状态：

```
[root@vm81 ~]# kubectl get node
NAME     STATUS    ROLES    AGE     VERSION
vm81     Ready     master   5d1h    v1.14.0
vm82     Ready     <none>    5d1h    v1.14.0
vm83     Ready     <none>    5d1h    v1.14.0
```

如果 node 没有处于 ready 状态，检查 pod：

```
[root@vm81 ~]# kubectl get pod -A
```

NAMESPACE	NAME	READY	STATUS	RESTARTS	AGE
kube-system	calico-kube-controllers-8dfd676d4-tfkms	1/1	Running	1	5d1h
kube-system	calico-node-8f6dl	1/1	Running	1	5d
kube-system	calico-node-tdgkv	1/1	Running	1	5d
kube-system	calico-node-zvwdz	1/1	Running	1	5d
kube-system	coredns-78498d8ff6-7f4dj	1/1	Running	1	5d1h
kube-system	coredns-78498d8ff6-nk54j	1/1	Running	1	5d1h
kube-system	etcd-vm81	1/1	Running	2	5d1h
kube-system	kube-apiserver-vm81	1/1	Running	2	5d1h
kube-system	kube-controller-manager-vm81	1/1	Running	2	5d1h
kube-system	kube-proxy-bnz6g	1/1	Running	2	5d1h
kube-system	kube-proxy-fvt6t	1/1	Running	2	5d1h
kube-system	kube-proxy-jw26k	1/1	Running	2	5d1h
kube-system	kube-scheduler-vm81	1/1	Running	2	5d1h
kube-system	kubernetes-dashboard-5957d4b56b-rjcm4	1/1	Running	1	5d1h

## 检查 pod

如果有的 pod 没有启动，检查 pod 的状态：

`kubectl describe pod pod-XXXXXXX`

## 查看日志

如果依然有问题，查看 kubelet 的日志，看问题针对性的解决：

```
[root@vm81 ~]# journalctl -f -u kubelet
-- Logs begin at 六 2020-08-01 14:42:27 CST. --
8月 01 14:43:06 vm81 kubelet[8553]: 2020-08-01 14:43:06.238
381f269cfd3937eff9fa80d97f2aeb4755597a7702be3585b931f085"
8月 01 14:43:06 vm81 kubelet[8553]: 2020-08-01 14:43:06.240
7eff9fa80d97f2aeb4755597a7702be3585b931f085" host="vm81"
8月 01 14:43:06 vm81 kubelet[8553]: 2020-08-01 14:43:06.240
81f269cfd3937eff9fa80d97f2aeb4755597a7702be3585b931f085" U
```

版权所有：有任何问题可以加我微信

