install kubernetes 1.7.5 on ubuntu 1604 (virtualbox vm)

写在前面：

K8s环境搭建是所有练习的第一步，也是所有练习的基础。

重点：

理解K8s的应用场景，基本组件和原理是本次练习的重点，也是为后面几个练习做准备，所以完成练习只是一个途径，更多的是要对K8s有一个比较完整的理解。下面会列出几个供学习和参考的网址，大家可以提前准备一下：

1. K8s官网地址：https://kubernetes.io/docs/home/
2. Jimmysong的K8s handbook：https://jimmysong.io/kubernetes-handbook/
3. Docker官网地址：https://www.docker.com/

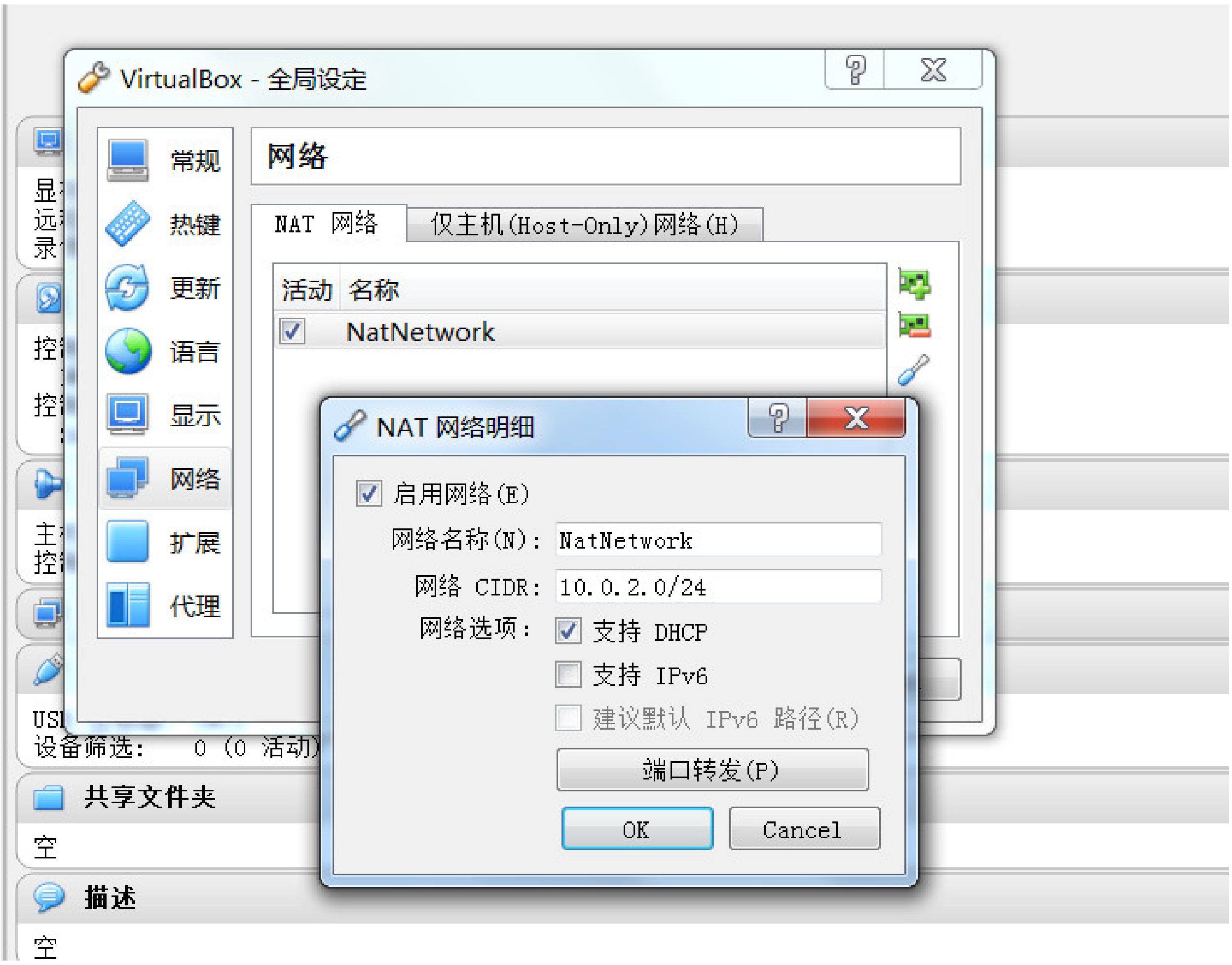
难点：

第一个练习本身并不难，更多的问题可能出现在虚拟机环境的准备上。下面列出几个可能碰到的问题。

坑1：网络配置

实验一本身不需要链接互联网，但是后续的实验需要进行联网下载镜像，所以建议在第一个实验时就配置好相关网络。配置过程如下：

在VirtualBox界面上点击"管理"->"全局设定"，在弹出框中选择"网络"->"NAT网络"，新建一个"NAT网络"，具体如下：



☑ NatNetwork

USB 设备筛选： 0 (0 活动)

共享文件夹

空

描述

空

NAT 网络明细

☑ 启用网络(E)

网络名称(N)： NatNetwork

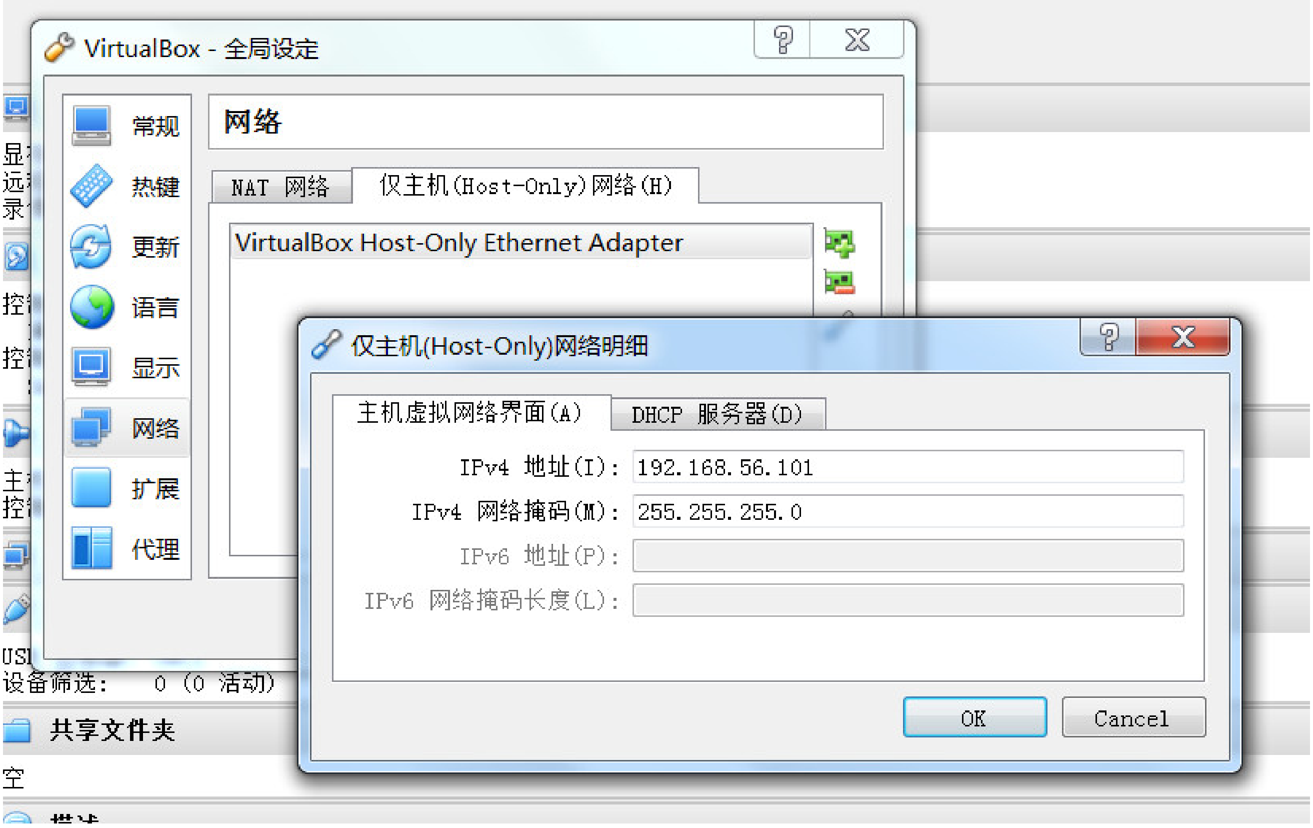
网络 CIDR： 10.0.2.0/24

网络选项： ☑ 支持 DHCP

□ 支持 IPv6

□ 建议默认 IPv6 路径(R)

选择"仅主机(Host-Only)网络(H)"，设置虚拟机内部网卡地址，如下：



VirtualBox Host-Only Ethernet Adapter

USB 设备筛选： 0 (0 活动)

空

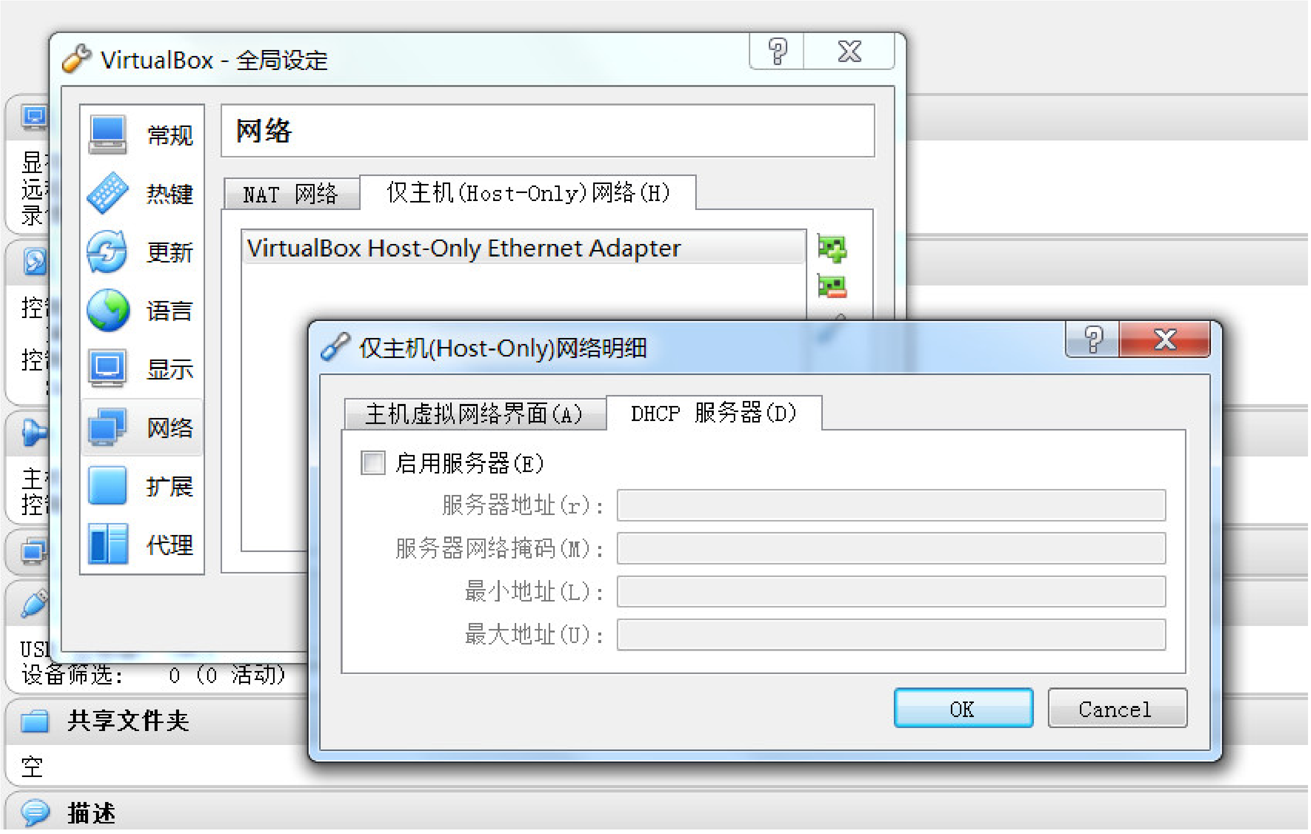
仅主机(Host-Only)网络明细

IPv4 地址(I)： 192.168.56.101

IPv4 网络掩码(M)： 255.255.255.0

IPv6 地址(P)：

IPv6 网络掩码长度(L)：



VirtualBox - 全局设定

网络

NAT 网络 | 仅主机(Host-Only)网络(H)

VirtualBox Host-Only Ethernet Adapter

仅主机(Host-Only)网络明细

主机虚拟网络界面(A) | DHCP 服务器(D)

□ 启用服务器(E)

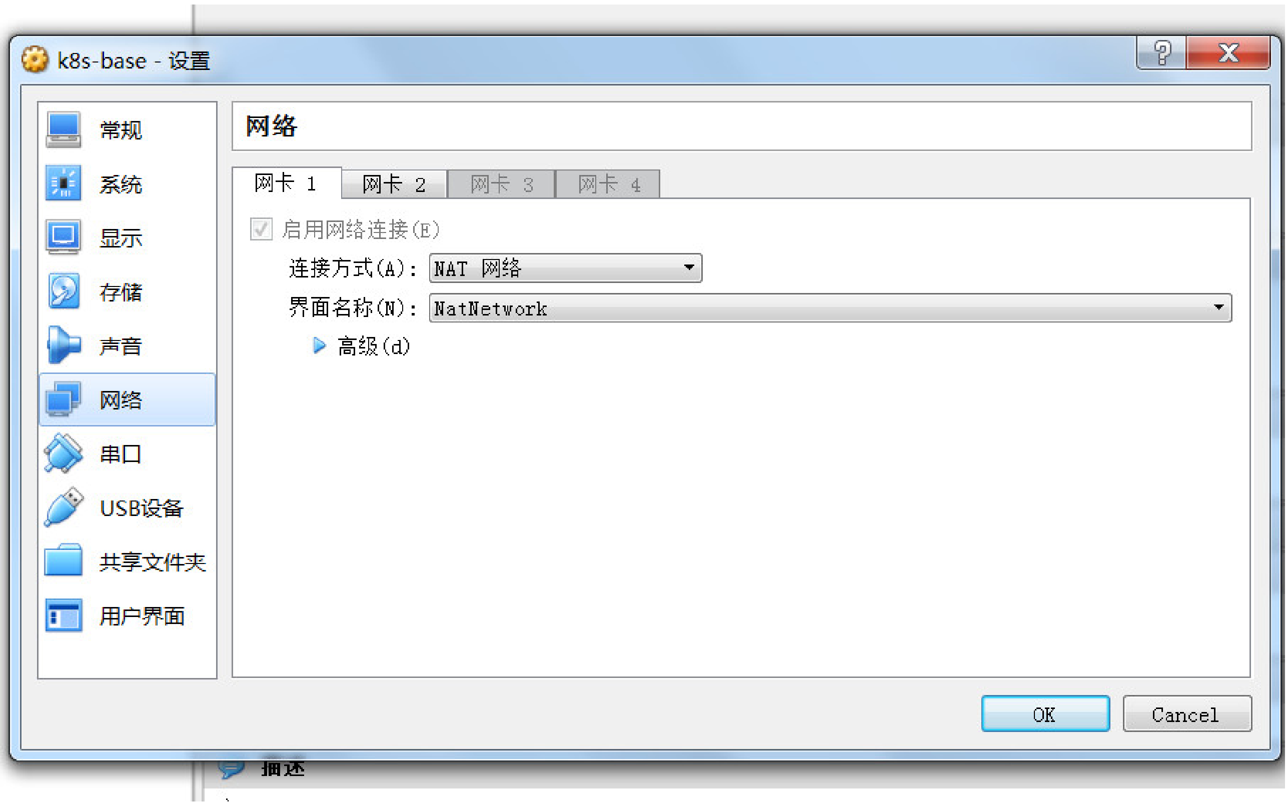
服务器地址(r):

服务器网络掩码(M):

最小地址(L):

最大地址(U):

右键虚拟机，选择"设置"->"网络"，新建"NAT 网络"和"仅主机(Host-Only)网络(H)"，如下：



k8s-base - 设置

网络

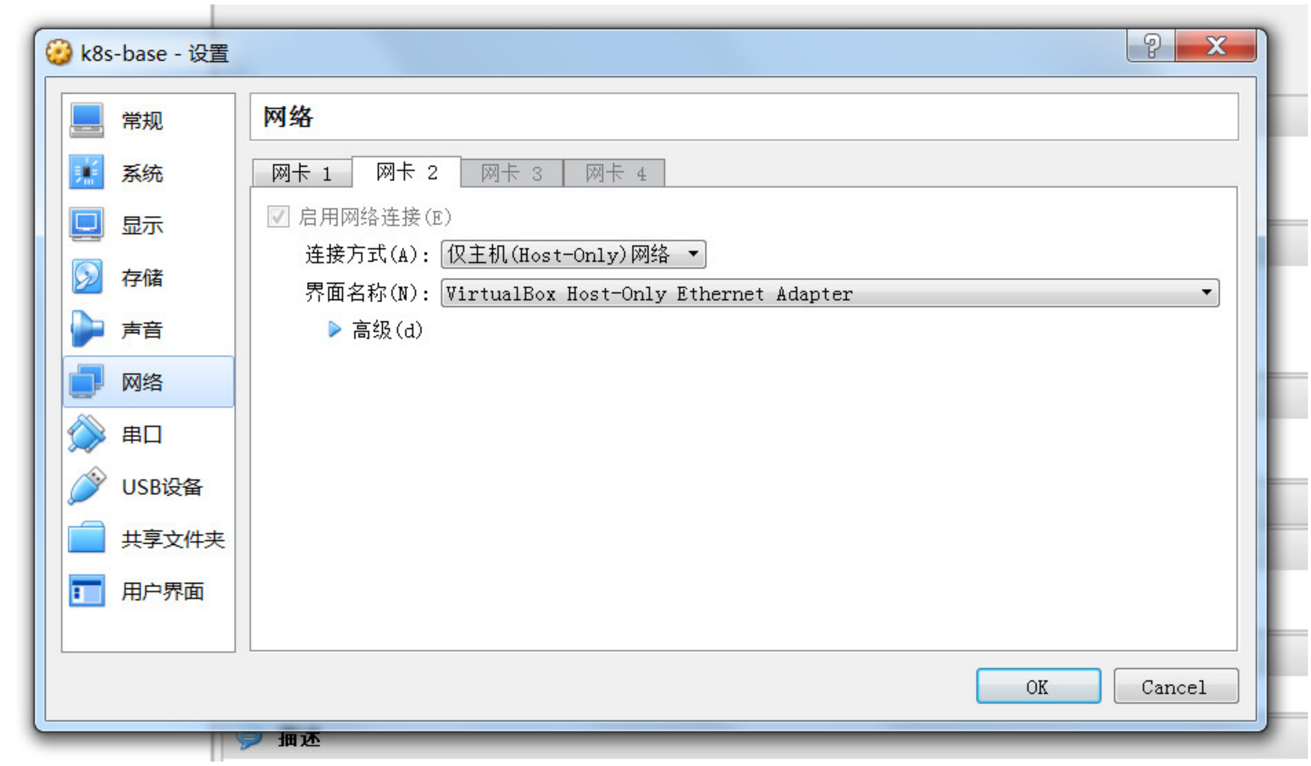
网卡 1 | 网卡 2 | 网卡 3 | 网卡 4

☑ 启用网络连接(E)

连接方式(A): NAT 网络

界面名称(N): NatNetwork

▶ 高级(d)



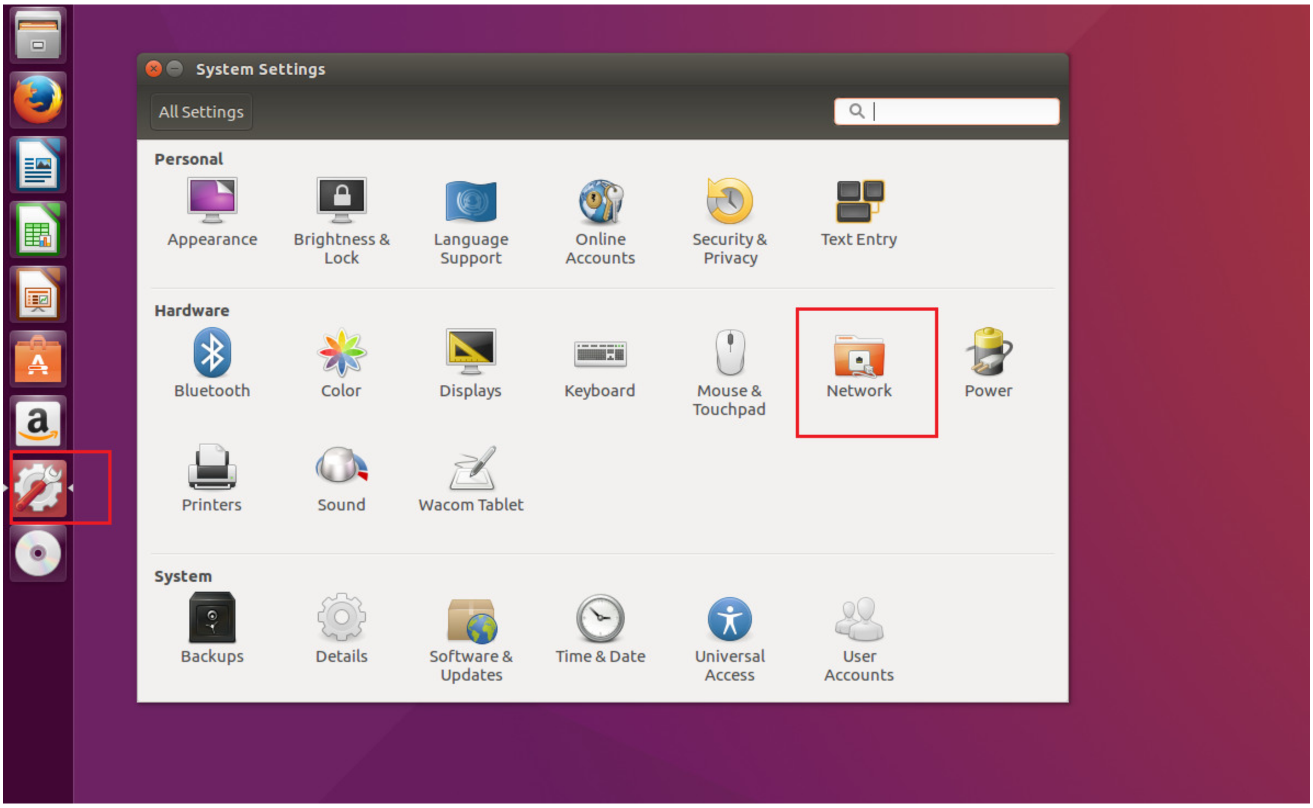
网络

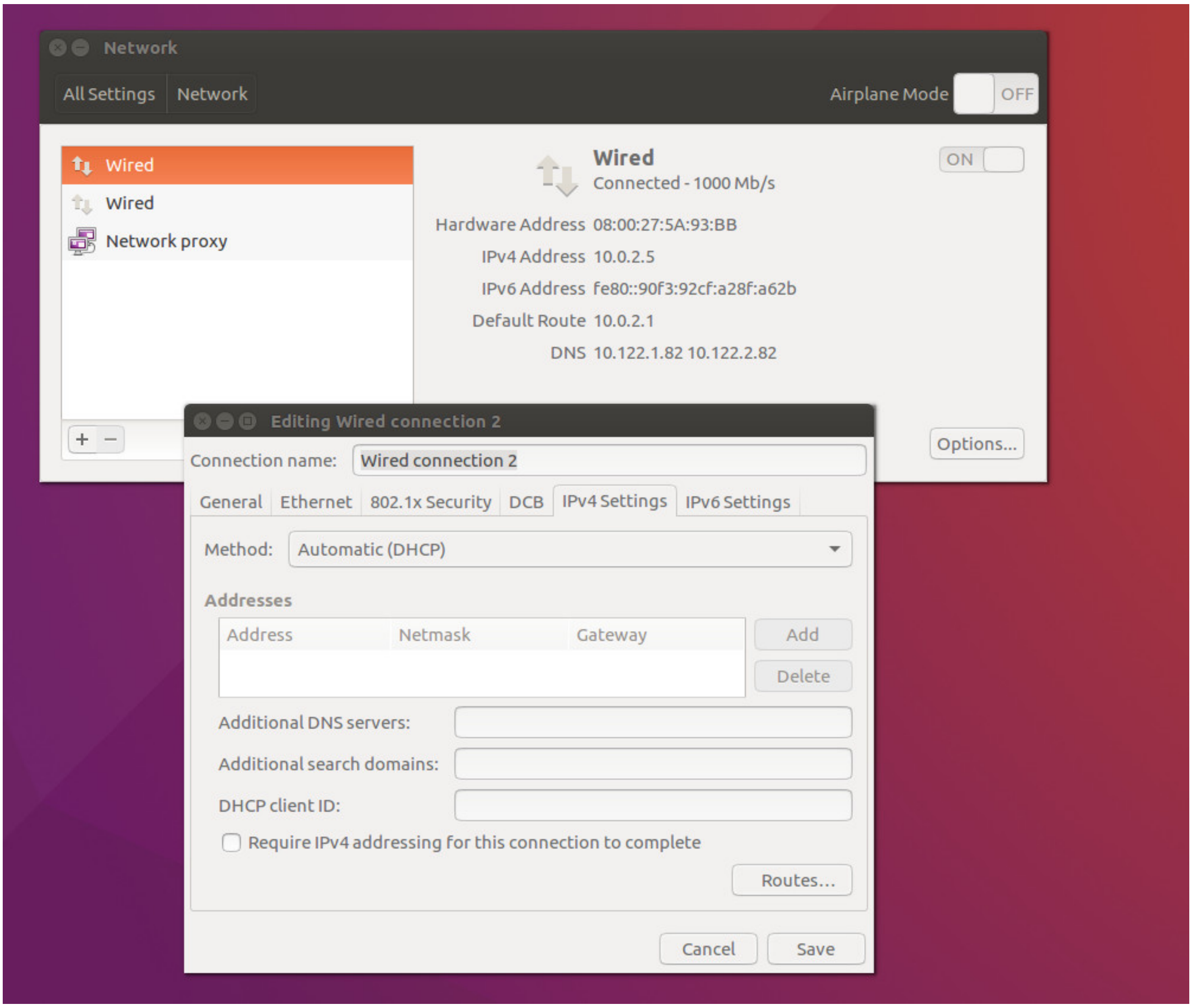
☑ 启用网络连接(E)

连接方式(A)： 仅主机(Host-Only)网络

界面名称(N)： VirtualBox Host-Only Ethernet Adapter

运行虚拟机，对网卡进行设置，如下：





Network

Wired
Connected - 1000 Mb/s

Hardware Address 08:00:27:5A:93:BB
IPv4 Address 10.0.2.5
IPv6 Address fe80::90f3:92cf:a28f:a62b
Default Route 10.0.2.1
DNS 10.122.1.82 10.122.2.82

Editing Wired connection 2

Connection name: Wired connection 2

Method: Automatic (DHCP)

Addresses

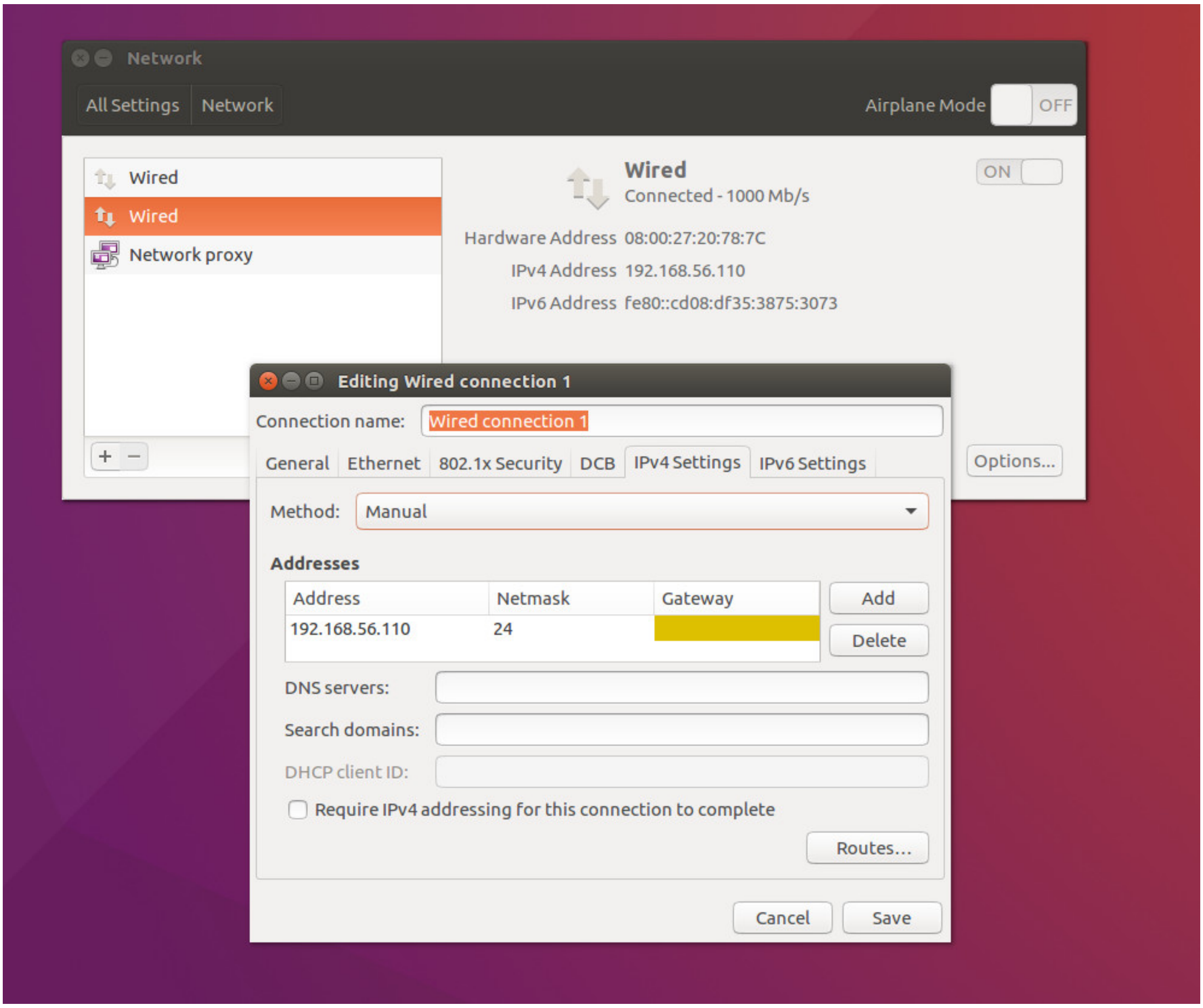
Address	Netmask	Gateway

Additional DNS servers:

Additional search domains:

DHCP client ID:

☐ Require IPv4 addressing for this connection to complete



Wired
Connected - 1000 Mb/s

Hardware Address 08:00:27:20:78:7C
IPv4 Address 192.168.56.110
IPv6 Address fe80::cd08:df35:3875:3073

Editing Wired connection 1

Connection name: Wired connection 1

Method: Manual

Addresses

Address	Netmask	Gateway
192.168.56.110	24	

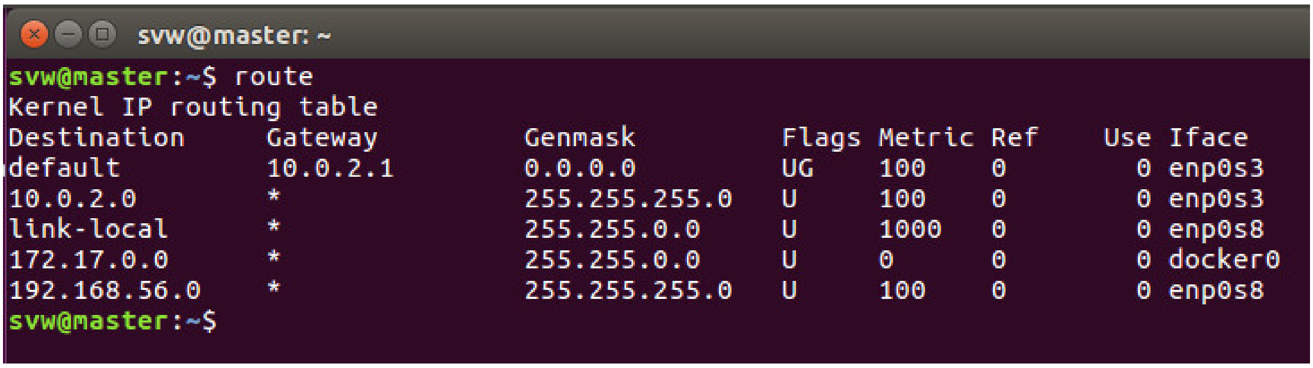
DNS servers:

Search domains:

DHCP client ID:

□ Require IPv4 addressing for this connection to complete

重启虚拟机，使用route命令，检查网卡设置是否正确：

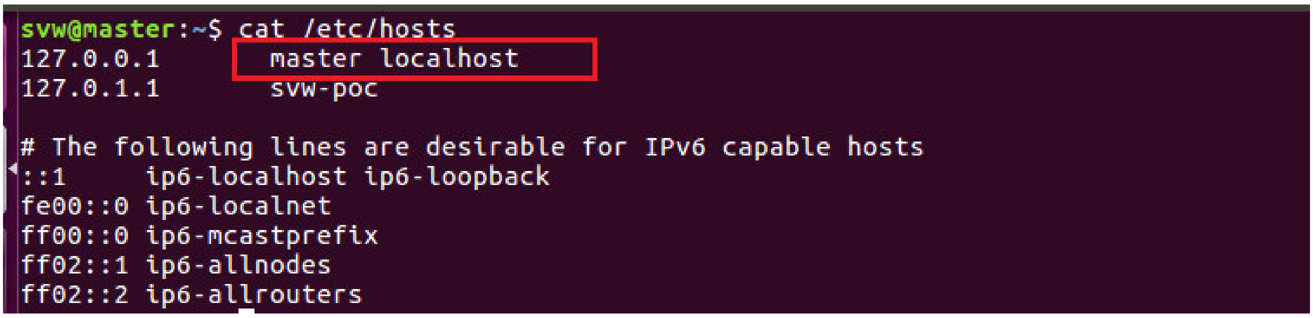


```
svw@master: ~
svw@master:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         10.0.2.1        0.0.0.0         UG    100    0        0 enp0s3
10.0.2.0        *               255.255.255.0   U     100    0        0 enp0s3
link-local      *               255.255.0.0     U     1000   0        0 enp0s8
172.17.0.0      *               255.255.0.0     U     0      0        0 docker0
192.168.56.0    *               255.255.255.0   U     100    0        0 enp0s8
svw@master:~$
```

使用ping命令，检查是否能够链接外网。

坑2：hosts文件配置

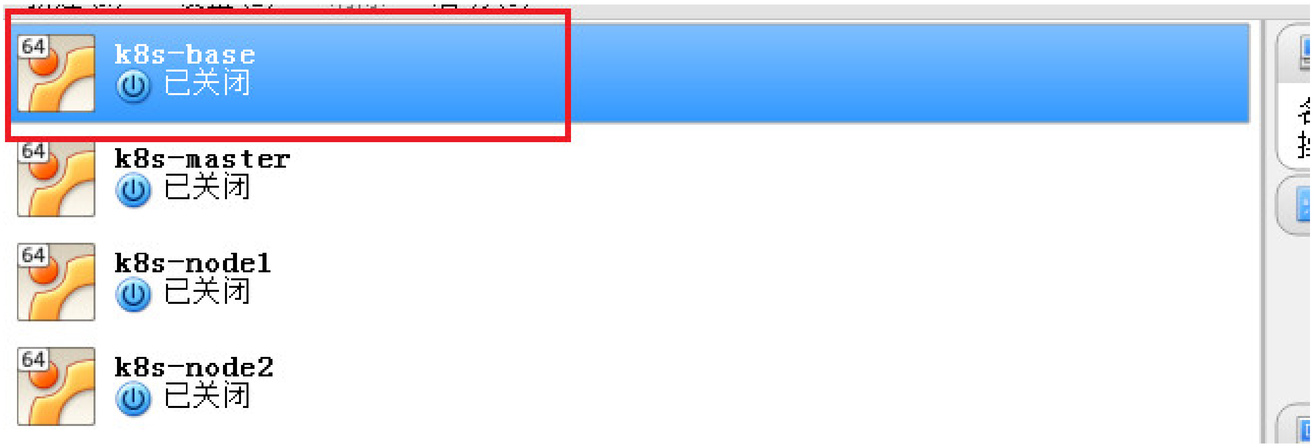
配置hosts文件时，要保留localhost的配置，否则会造成master或者node节点启动失败，如下：



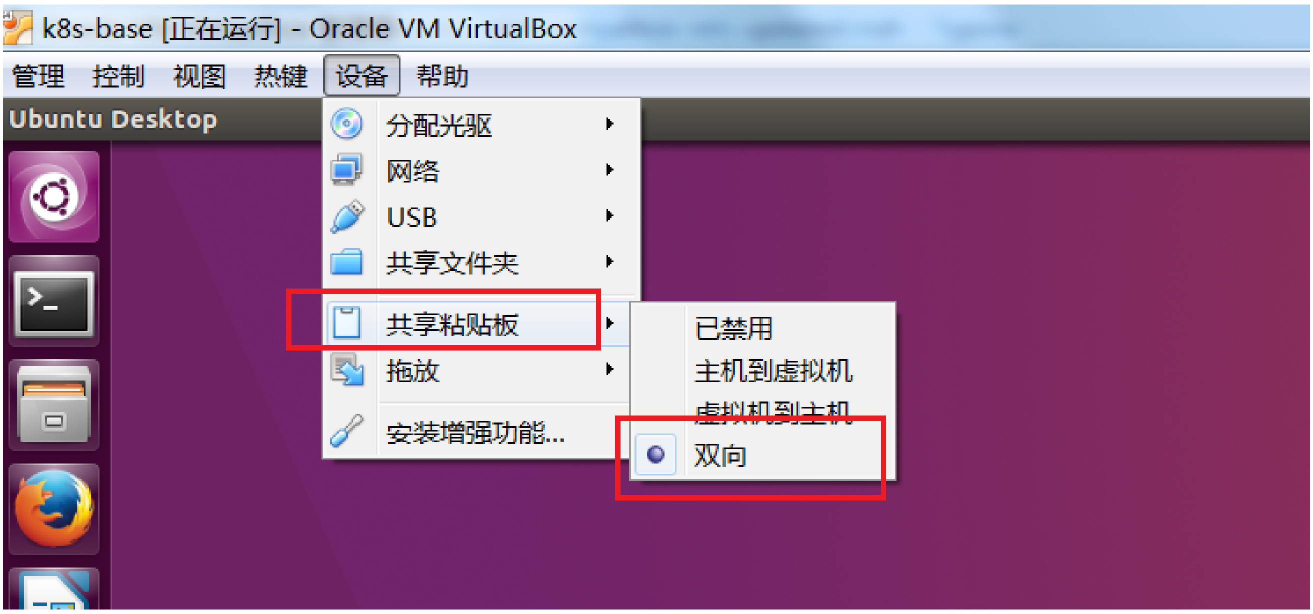
```
svw@master:~$ cat /etc/hosts
127.0.0.1       master localhost
127.0.1.1       svw-poc

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

建议：

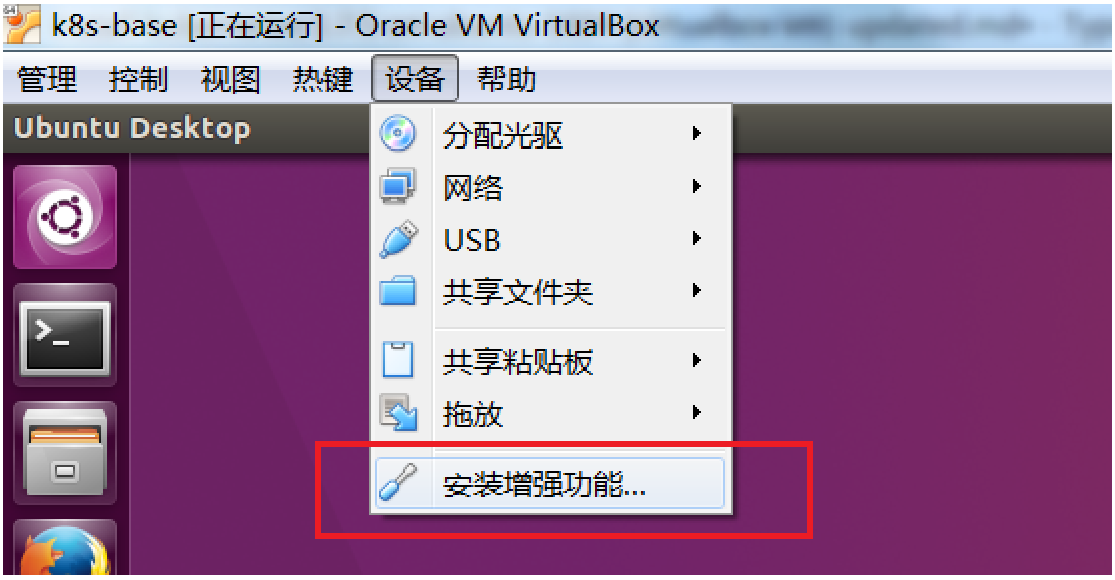
建议1：创建一个基础虚拟机，安装基本功能和设置，其它的虚拟机从这个基础虚拟机复制后，再进行练习，如下：



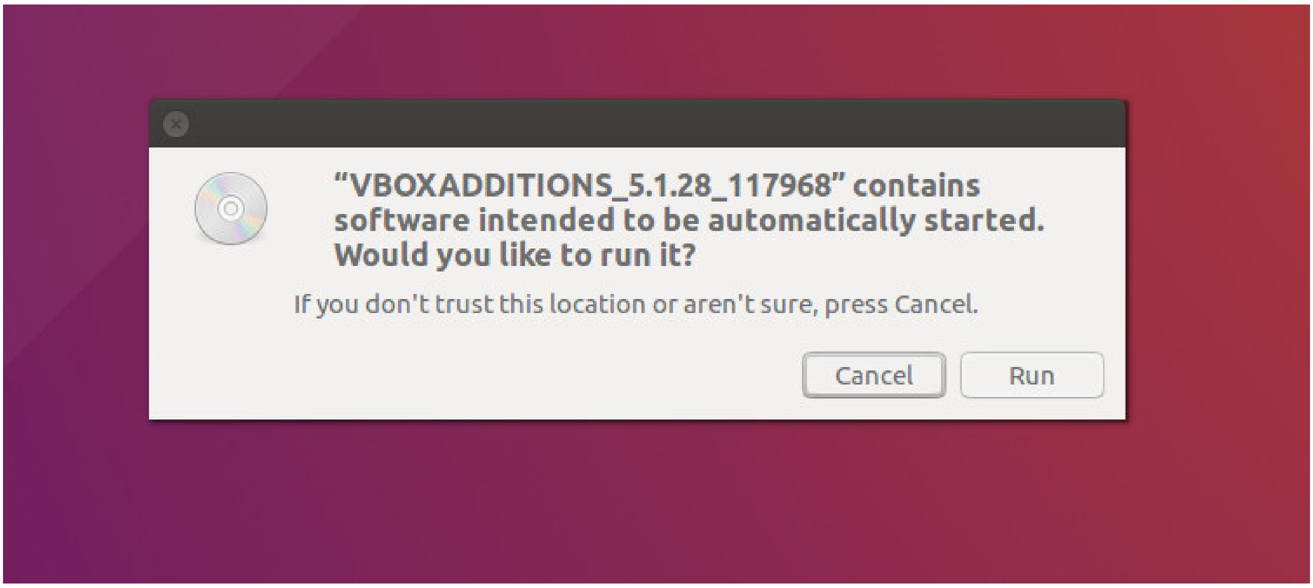
K8s-base中安装与宿主机进行同步拷贝，粘贴功能：



安装增强功能：



点击“Run”，输入管理员密码“svw”，并等待安装完成：



“VBOXADDITIONS_5.1.28_117968” contains software intended to be automatically started. Would you like to run it?

If you don't trust this location or aren't sure, press Cancel.

宿主机环境要求

- VirtualBox 5.1.22以上版本
- 虚拟机网络启用virtualbox host-only模式，网卡IP地址设定为192.168.56.1

虚拟机模板说明

操作系统版本：ubuntu 16.04

账号：svw/svw

虚拟机网卡：单网卡，host-only模式；注意务必不要启用NAT网卡，那会造成flannel网络组件获取地址错误。

虚拟机内已安装组件清单：ssh, curl, docker, kublet, kubeadm, kubectl

安装master节点

从模板复制虚拟机

基于提供的模板虚拟机复制出一个新的虚拟机（注意务必选中 重新初始化所有的网卡MAC地址）

设置静态IP地址

```
sudo vi  /etc/network/interfaces
```

将以下内容添加到文件中保存退出。

```
# The primary network interface
auto enp0s3

iface enp0s3 inet static
address 192.168.56.110
netmask 255.255.255.0
gateway 192.168.56.1
```

更改主机名

```
sudo vi /etc/hostname
```

将主机名改为master， 保存退出

更改hosts文件

```
sudo vi /etc/hosts
```

将127.0.0.1对应的主机名改为master，保存退出

重启服务器

```
sudo init 6
```

导入镜像

将K8s所需要的镜像(google_containers.tar)上转至服务器目录/home/svw（通过winscp上传）

```
 sudo docker load -i /home/svw/google_containers.tar
```

查看下导入的镜像列表

```
svw@master:~$ sudo docker images
REPOSITORY                                               TAG                 IMAGE ID
CREATED             SIZE
gcr.io/google_containers/kube-apiserver-amd64            v1.7.5              33cecf00bef1
 3 weeks ago         186.1 MB
gcr.io/google_containers/kube-controller-manager-amd64   v1.7.5              4d581c01252d
 3 weeks ago         138 MB
gcr.io/google_containers/kube-scheduler-amd64            v1.7.5              2aad6febbb16
 3 weeks ago         77.2 MB
gcr.io/google_containers/kube-proxy-amd64                v1.7.5              1314eb3ac430
 3 weeks ago         114.7 MB
quay.io/coreos/flannel                                   v0.8.0-amd64        9db3bab8c19e
 11 weeks ago        50.73 MB
gcr.io/google_containers/k8s-dns-sidecar-amd64           1.14.4              38bac66034a6
 3 months ago        41.82 MB
gcr.io/google_containers/k8s-dns-kube-dns-amd64          1.14.4              a8e00546bcf3
 3 months ago        49.39 MB
gcr.io/google_containers/k8s-dns-dnsmasq-nanny-amd64     1.14.4              f7f45b9cb733
 3 months ago        41.42 MB
gcr.io/google_containers/etcd-amd64                      3.0.17              243830dae7dd
 7 months ago        168.9 MB
gcr.io/google_containers/pause-amd64                     3.0                 99e59f495ffa
 17 months ago       746.9 kB
```

安装maseter上的K8s组件

```
sudo kubeadm init --kubernetes-version v1.7.5 --apiserver-advertise-address 192.168.56.110 --
pod-network-cidr 10.244.0.0/16
```

其中，`--apiserver-advertise-address` 对应的值为master节点的本机IP地址，`--pod-network-cidr`对应的值为k8s群集内部的网络段。

运行成功后会有以下内容打印出

```
svw@master:~$ kubeadm init --kubernetes-version v1.7.5 --apiserver-advertise-address
192.168.56.110 --pod-network-cidr 10.244.0.0/16
[kubeadm] WARNING: kubeadm is in beta, please do not use it for production clusters.
[init] Using Kubernetes version: v1.7.5
[init] Using Authorization modes: [Node RBAC]
[preflight] Running pre-flight checks
[preflight] Some fatal errors occurred:
        user is not running as root
[preflight] If you know what you are doing, you can skip pre-flight checks with `--skip-
preflight-checks`
svw@master:~$ sudo kubeadm init --kubernetes-version v1.7.5 --apiserver-advertise-address
192.168.56.110 --pod-network-cidr 10.244.0.0/16
[kubeadm] WARNING: kubeadm is in beta, please do not use it for production clusters.
[init] Using Kubernetes version: v1.7.5
[init] Using Authorization modes: [Node RBAC]
[preflight] Running pre-flight checks

[kubeadm] WARNING: starting in 1.8, tokens expire after 24 hours by default (if you require a
non-expiring token use --token-ttl 0)
[certificates] Generated CA certificate and key.
[certificates] Generated API server certificate and key.
[certificates] API Server serving cert is signed for DNS names [master kubernetes
kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs
[10.96.0.1 192.168.56.110]
[certificates] Generated API server kubelet client certificate and key.
[certificates] Generated service account token signing key and public key.
[certificates] Generated front-proxy CA certificate and key.
[certificates] Generated front-proxy client certificate and key.
[certificates] Valid certificates and keys now exist in "/etc/kubernetes/pki"
[kubeconfig] Wrote KubeConfig file to disk: "/etc/kubernetes/kubelet.conf"
[kubeconfig] Wrote KubeConfig file to disk: "/etc/kubernetes/controller-manager.conf"
[kubeconfig] Wrote KubeConfig file to disk: "/etc/kubernetes/scheduler.conf"
[kubeconfig] Wrote KubeConfig file to disk: "/etc/kubernetes/admin.conf"
[apiclient] Created API client, waiting for the control plane to become ready
[apiclient] All control plane components are healthy after 36.500720 seconds
[token] Using token: e99d39.674b685c4270a1d7
[apiconfig] Created RBAC rules
[addons] Applied essential addon: kube-proxy
[addons] Applied essential addon: kube-dns

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 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 e99d39.674b685c4270a1d7 192.168.56.110:6443
```

注意：最后一句需要保存好，未来安装新的node节点时都会用到。

配置kubectl，在svw用户环境下逐条执行以下命令

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

执行kubectl检查各组件的状态，正常情况下会有以下内容输出

```
svw@master:~$ kubectl get all --all-namespaces
NAMESPACE     NAME                                 READY     STATUS    RESTARTS   AGE
kube-system   po/etcd-master                       1/1       Running   0          2m
```

```
kube-system   po/kube-apiserver-master            1/1       Running   0          2m
kube-system   po/kube-controller-manager-master   1/1       Running   0          2m
kube-system   po/kube-dns-2425271678-hmblq        0/3       Pending   0          3m
kube-system   po/kube-proxy-cp11m                 1/1       Running   0          3m
kube-system   po/kube-scheduler-master            1/1       Running   0          2m

NAMESPACE     NAME             CLUSTER-IP   EXTERNAL-IP   PORT(S)         AGE
default       svc/kubernetes   10.96.0.1    <none>        443/TCP         3m
kube-system   svc/kube-dns     10.96.0.10   <none>        53/UDP,53/TCP   3m

NAMESPACE     NAME              DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
kube-system   deploy/kube-dns   1         1         1            0           3m

NAMESPACE     NAME                     DESIRED   CURRENT   READY     AGE
kube-system   rs/kube-dns-2425271678   1         1         0         3m
```

注：dns组件状态为pending是因为网络组件还未安装完成，属正常情况。

安装网络组件flannel

将kube-flannel.yml、kube-flannel-rbac.yml上传至/home/svw

执行安装命令

```
kubectl apply -f kube-flannel-rbac.yml
kubectl apply -f kube-flannel.yml
```

执行完成后等待两分钟后会有如下输出， master节点安装完成。

```
svw@master:~$ kubectl get all --all-namespaces
NAMESPACE     NAME                                READY     STATUS    RESTARTS   AGE
kube-system   po/etcd-master                      1/1       Running   0          2m
kube-system   po/kube-apiserver-master            1/1       Running   0          2m
kube-system   po/kube-controller-manager-master   1/1       Running   0          2m
kube-system   po/kube-dns-2425271678-hmblq        3/3       Running   0          7m
kube-system   po/kube-flannel-ds-rqp7j            2/2       Running   0          42s
kube-system   po/kube-proxy-cp11m                 1/1       Running   0          7m
kube-system   po/kube-scheduler-master            1/1       Running   0          2m

NAMESPACE     NAME             CLUSTER-IP   EXTERNAL-IP   PORT(S)         AGE
default       svc/kubernetes   10.96.0.1    <none>        443/TCP         7m
kube-system   svc/kube-dns     10.96.0.10   <none>        53/UDP,53/TCP   7m

NAMESPACE     NAME              DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
kube-system   deploy/kube-dns   1         1         1            1           7m

NAMESPACE     NAME                     DESIRED   CURRENT   READY     AGE
kube-system   rs/kube-dns-2425271678   1         1         1         7m
```

安装node1节点

从模板复制虚拟机

基于提供的模板虚拟机复制出一个新的虚拟机（注意务必选中 重新初始化所有的网卡MAC地址）

设置静态IP地址

```
sudo vi  /etc/network/interfaces
```

将以下内容添加到文件中保存退出。

```
# The primary network interface
auto enp0s3

iface enp0s3 inet static
address 192.168.56.111
netmask 255.255.255.0
gateway 192.168.56.1
```

更改主机名

```
sudo vi /etc/hostname
```

将主机名改为node1， 保存退出

更改hosts文件

```
sudo vi /etc/hosts
```

将127.0.0.1对应的主机名改为node1，保存退出

重启服务器

```
sudo init 6
```

导入镜像

将K8s所需要的镜像(google_containers.tar)上转至服务器目录/home/svw（通过winscp上传）

```
 sudo docker load -i /home/svw/google_containers.tar
```

将node1加入到master节点

执行以下命令将node1加入到master节点

```
sudo kubeadm join --token e99d39.674b685c4270a1d7 192.168.56.110:6443
```

执行成功会得到以下输出

```
svw@node1:~$ sudo kubeadm join --token e99d39.674b685c4270a1d7 192.168.56.110:6443
```

```
[kubeadm] WARNING: kubeadm is in beta, please do not use it for production clusters.
[preflight] Running pre-flight checks
[discovery] Trying to connect to API Server "192.168.56.110:6443"
[discovery] Created cluster-info discovery client, requesting info from
"https://192.168.56.110:6443"
[discovery] Cluster info signature and contents are valid, will use API Server
"https://192.168.56.110:6443"
[discovery] Successfully established connection with API Server "192.168.56.110:6443"
[bootstrap] Detected server version: v1.7.5
[bootstrap] The server supports the Certificates API (certificates.k8s.io/v1beta1)
[csr] Created API client to obtain unique certificate for this node, generating keys and
certificate signing request
[csr] Received signed certificate from the API server, generating KubeConfig...
[kubeconfig] Wrote KubeConfig file to disk: "/etc/kubernetes/kubelet.conf"

Node join complete:
* Certificate signing request sent to master and response
  received.
* Kubelet informed of new secure connection details.

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

master节点上查看整个群集的状态

```
svw@master:~$ kubectl get nodes
NAME      STATUS    AGE       VERSION
master    Ready     1h        v1.7.5
node1     Ready     31s       v1.7.5
svw@master:~$ kubectl get all --all-namespaces
NAMESPACE     NAME                                READY     STATUS    RESTARTS   AGE
kube-system   po/etcd-master                      1/1       Running   0          1h
kube-system   po/kube-apiserver-master            1/1       Running   0          1h
kube-system   po/kube-controller-manager-master   1/1       Running   0          1h
kube-system   po/kube-dns-2425271678-hmblq        3/3       Running   0          1h
kube-system   po/kube-flannel-ds-55kfm            2/2       Running   0          34s
kube-system   po/kube-flannel-ds-rqp7j            2/2       Running   0          1h
kube-system   po/kube-proxy-04fbb                 1/1       Running   0          34s
kube-system   po/kube-proxy-cp11m                 1/1       Running   0          1h
kube-system   po/kube-scheduler-master            1/1       Running   0          1h

NAMESPACE     NAME             CLUSTER-IP   EXTERNAL-IP   PORT(S)         AGE
default       svc/kubernetes   10.96.0.1    <none>        443/TCP         1h
kube-system   svc/kube-dns     10.96.0.10   <none>        53/UDP,53/TCP   1h

NAMESPACE     NAME              DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
kube-system   deploy/kube-dns   1         1         1            1           1h

NAMESPACE     NAME                     DESIRED   CURRENT   READY     AGE
kube-system   rs/kube-dns-2425271678   1         1         1         1h
```