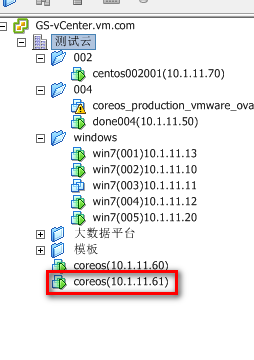
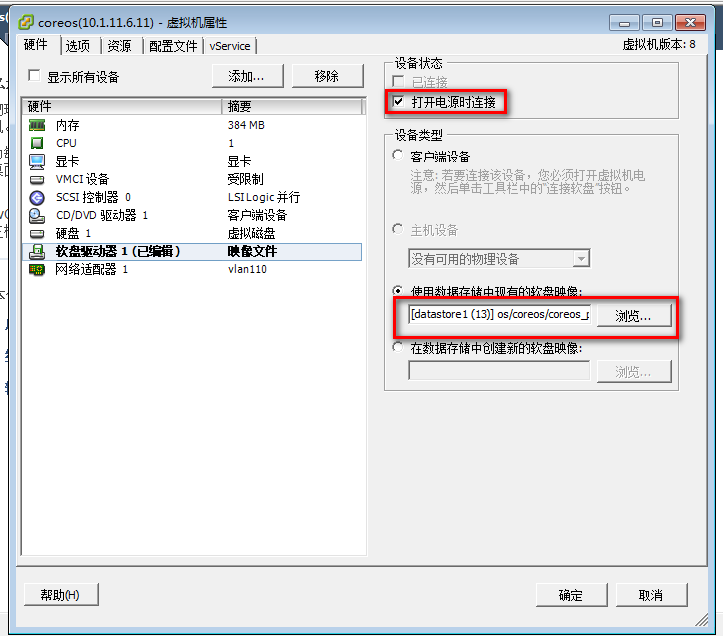
## CoreOS安装步骤

### ca1、创建虚拟机

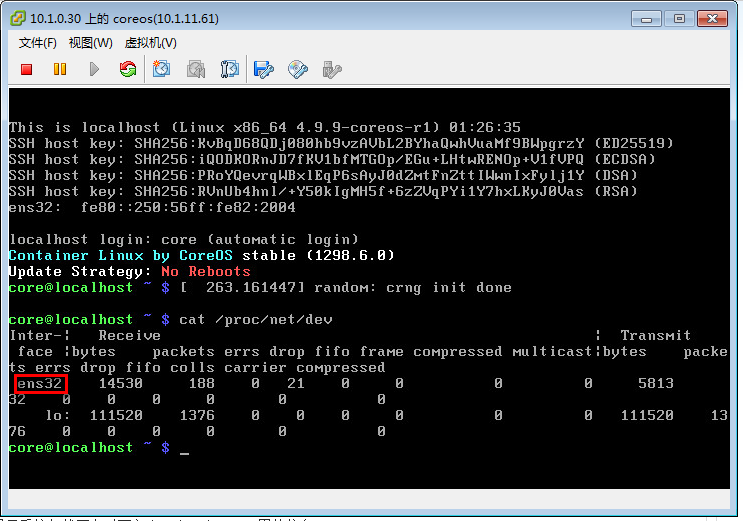
vCenter中创建虚拟机，在其名称中先指定要设定的IP地址，虚拟机设置中光驱中挂载下载的coreos的iso的文件。其下载地址为：<https://stable.release.core-os.net/amd64-usr/current/coreos_production_iso_image.iso>。



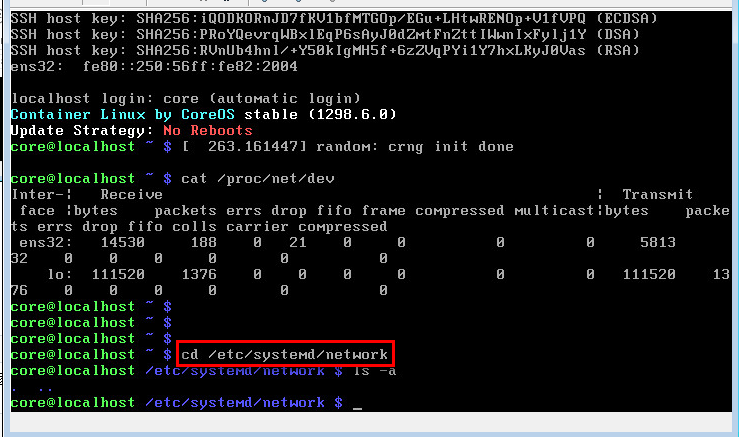


### s2、配置CD Live的网络

1、进入虚拟机，通过 cat /proc/net/dev 查看正在使用的网卡名，在配置网络时要使用到该名称。

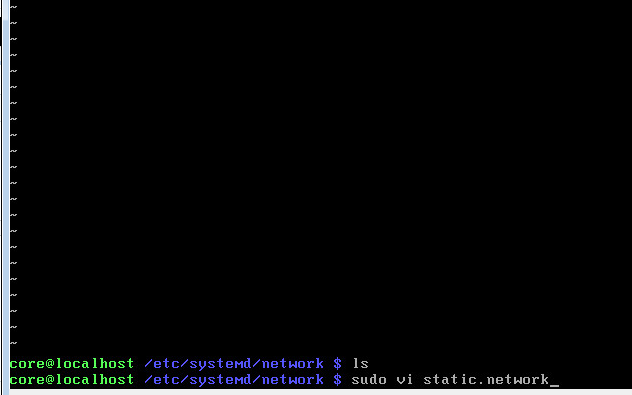


2、进入目录/etc/systemd/network/cat



3、通过vi来创建static.network进行网络的配置。

sudo vi **static**.network



输入i，进行vi的insert模式， 填入如下内容，你需要根据你实际情况修改

cat /proc/net/dev

/etc/systemd/network

wget http://10.1.11.20/cc61.yml

wget <http://10.1.11.20/cloud-config.yaml>

sudo coreos-install -d /dev/sda -C stable -c cc61.yaml -b <http://10.1.11.20>

vi cloud-config.yaml

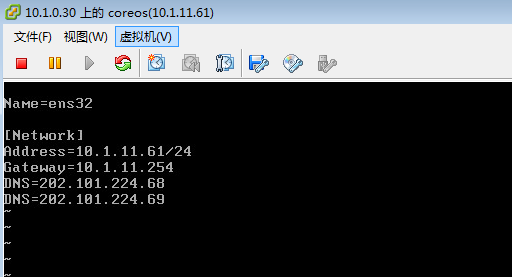
sudo vi **static**.network

sudo systemctl restart systemd-networkd

sudo coreos-install -d /dev/sda -C stable -c cloud-config.yaml -b <http://10.1.11.20>

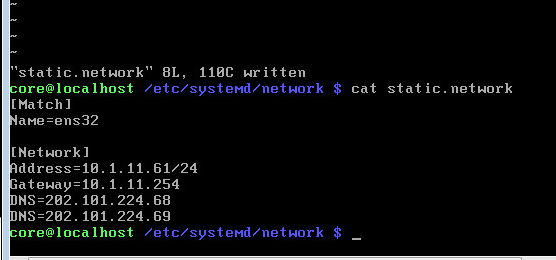
sudo coreos-install -d /dev/sda -C stable -c cc82.yml -b http://10.1.11.20

|  |
| --- |
| [Match]  Name=ens32 #网卡名 ，上面查询得到的名称    [Network]  Address=10.1.11.61/24   #需要设定的ip  Gateway=10.1.11.254   #网关  DNS=202.101.224.68 #没有DNS，通过域名访问不了网络   DNS=202.101.224.69 |



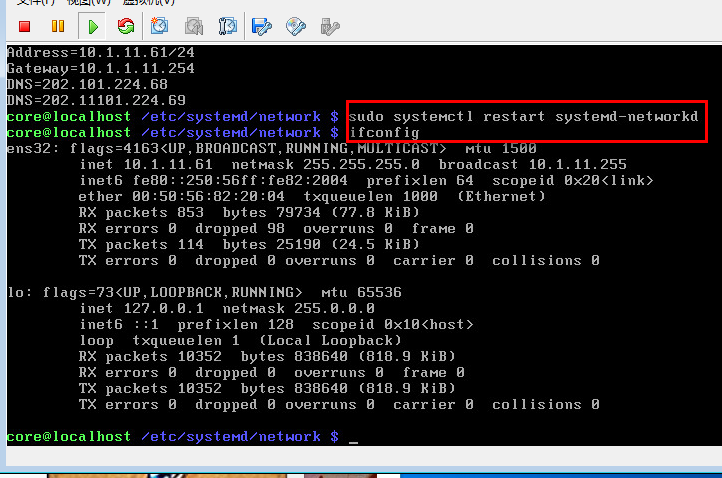
先按esc键，输入：，输入wq，即保存退出。

4、通过cat查看一下该文件的内容。



5、使得网络启效，使用下面命令：sudo systemctl restart systemd-networkd

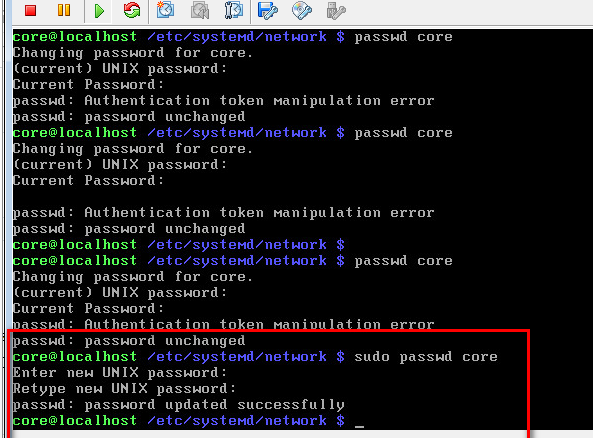
接下来通过ifconfig查看一下是否启作用，可以通过ping网关看网络是否通了。



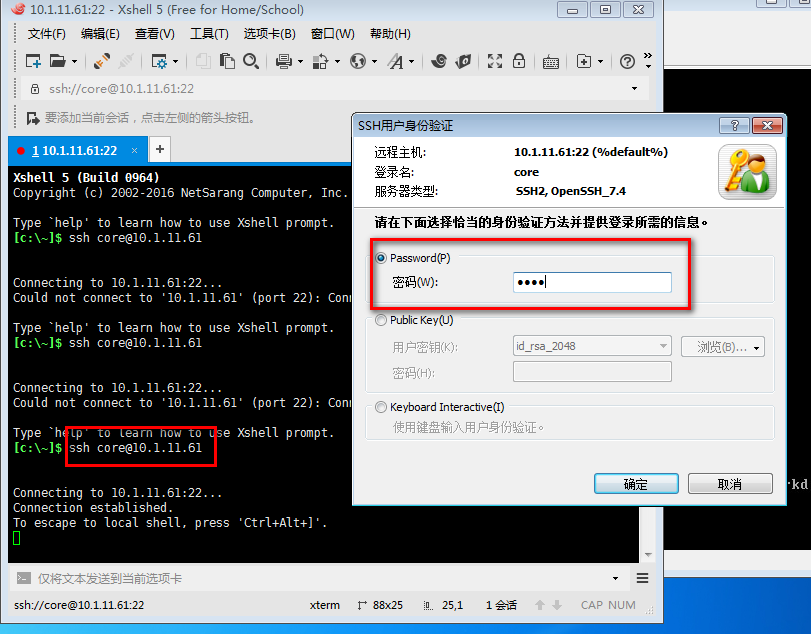
3、配置SSH登录

Coreos安装时需要进行一些配置，直接在虚拟机中操作不方便，最好配置SSH登录到coreos虚拟机中进行操作。

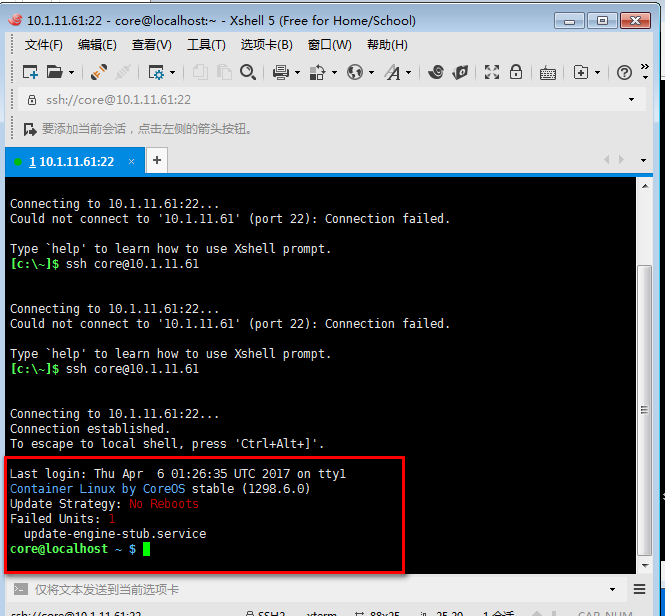
1、修改core的用户的密码，网络安装说明的都是采用root，coreos已经把root给禁了，只能采用core的用户。密码设定为:core或随意。



2、ssh登录，如果网络是通的话，通过ssh一般都可以登录，在Mac中直接使用命令终端，在windows下可以使用Xshell工具来进行操作。



登录成功之后就可以在ssh的客户端进行操作，可以通过copy，paste等方便编写配置文件。

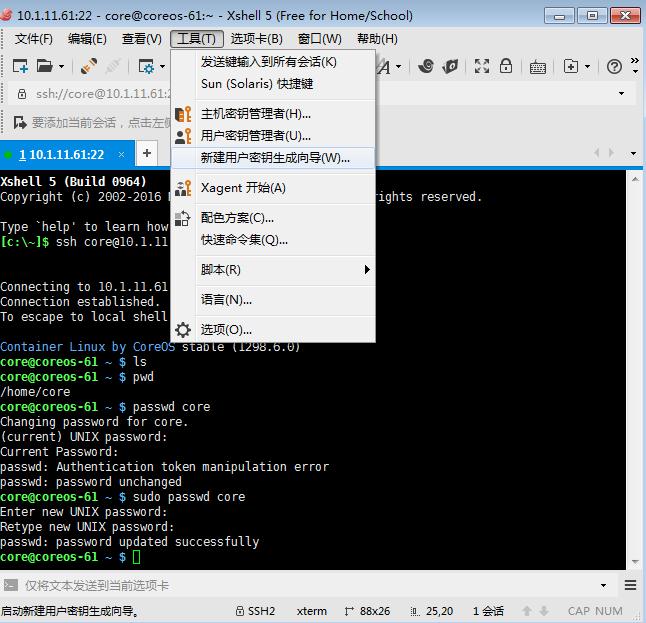


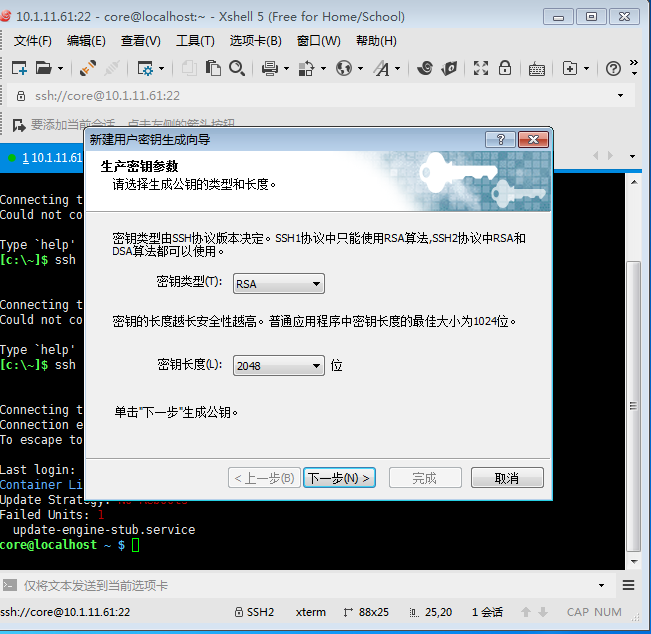
### wvi4、配置cloud-config.yaml

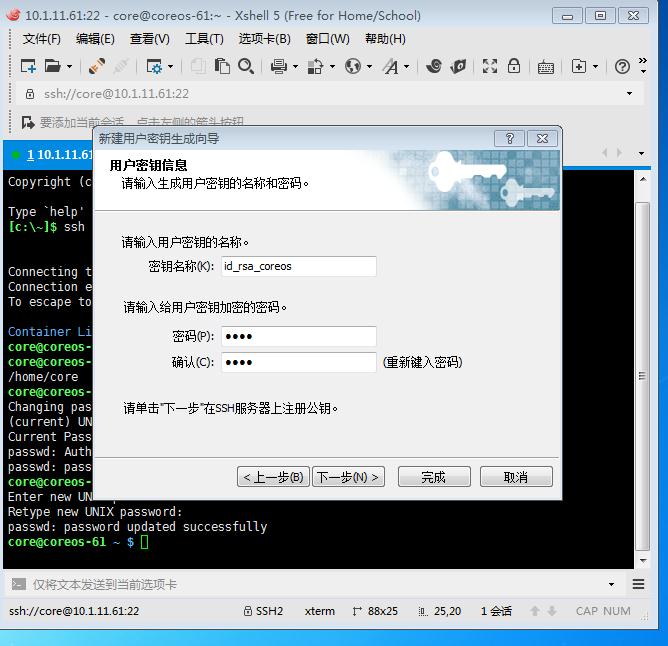
cloud-config.yaml是把coreos从cd live安装到硬盘上的配置文件，目前只需要配置其网络和ssh免登录的公钥。其网络和第二步的网络配置和相同。公钥需要创建。下表中红色的需要修改为自己的网络配置和公钥。

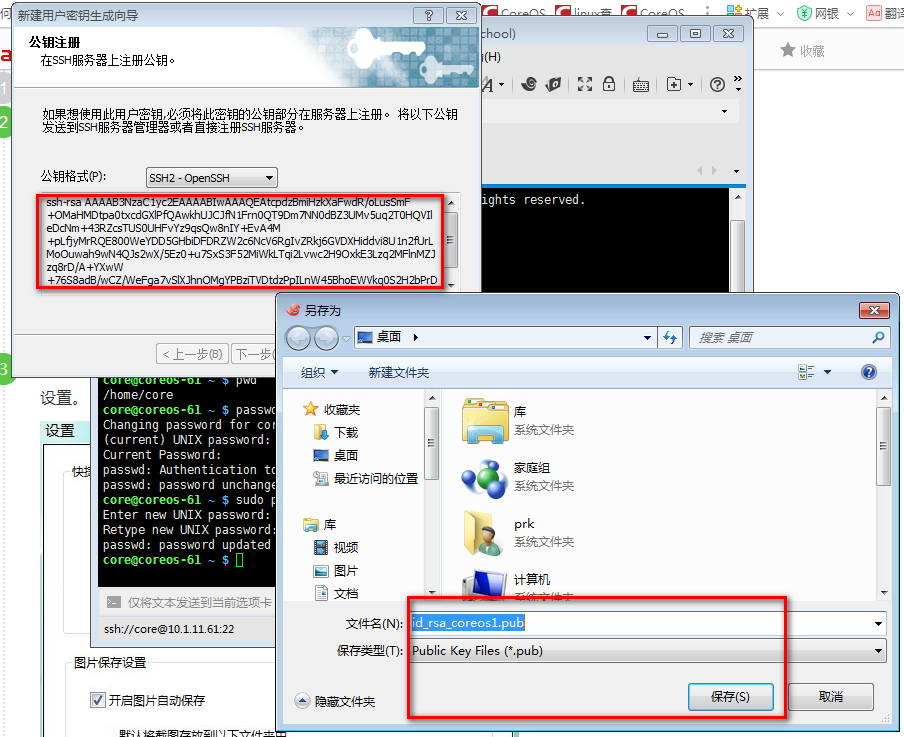
|  |
| --- |
| #cloud-config  hostname: coreos-61  coreos:  etcd:  addr: $private\_ipv4:4001  peer-addr: $private\_ipv4:7001  units:  - name: etcd.service  command: start  - name: fleet.service  command: start  - name: static.network  content: |  [Match]  Name=ens32  [Network]  Address=10.1.11.61/24  Gateway=10.1.11.254  DNS=202.101.224.68  DNS=202.101.224.69  users:  - name: core  ssh-authorized-keys:  - ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAq1ic0xd0eRx3A/33EpHu55bdw14hnbwi3yjh/cZ/hsF0x91pEhwDes6BsYdZGvF33uyWYQN5JEk24/t+USVfbTbDFH+qBYMwlnKxT27/RnILwsNUAHwdik0SeAs4Z9MuXcaZtoCz+CpRf3le7zBbegstp5k/bQoyhsGbplu3v9kJh1IWizXTWgx8N7ldFtYsPjzKCSYOavm88zyadlf24hwWNcqPH8KxGUFptG4Ru1y604op52xqsB/nv7+M8xANZRjuKP4Iqa1ywc/bOhAP28IpukEHMVqDtQphfRMsiXY9rxmy6Py9NztQzg0Gezwmns1fpG7efauEaqEQ+tixwQ==  - groups:  - sudo  - docker |

1、新建ssh公钥和密钥。在mac中采用ssh-keygen命令生成。这里采用xshell工具生成。



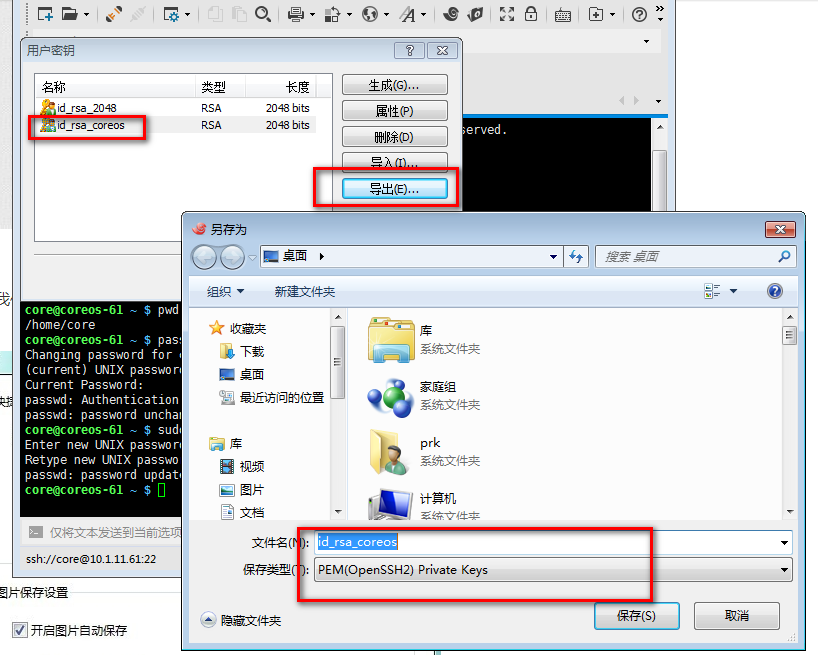


这里设定了密码在后面登录时需要使用，可以不设定，最好设定。



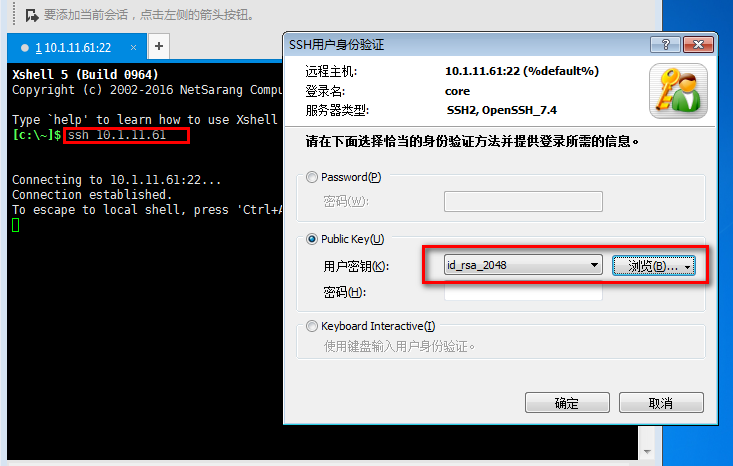
本图中的生成公钥把其内容复制到cloud-config.yaml文件其公钥的位置上。也可以把其文件保存下来。

2、导出私钥

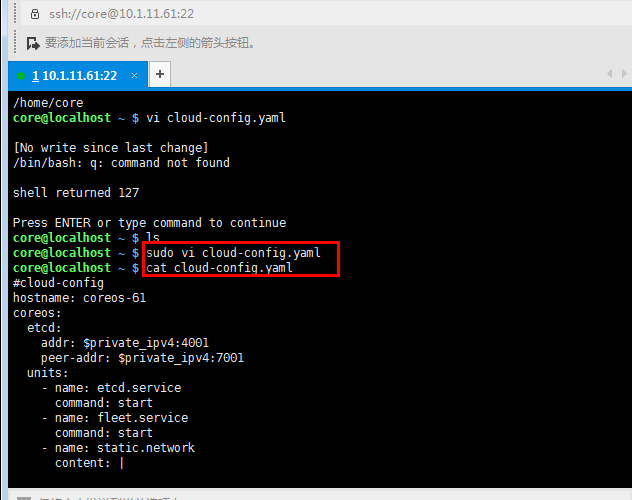


先选择公钥，然后导出按钮，保存。

3、在安装完成之后（本文档的第五步完成），先ssh [core@10.1.11.61](mailto:core@10.1.11.61),然后在弹出的对话中选择保存的私钥。



3、在ssh客户端(Xshell)中，先在core用户的当前目录下通过vi创建cloud-config.yaml文件，然后把整理好的cloud-config.yaml的内容复制到进来，保存，并通过cat查看。



【可选方案】也可以把cloud-config.yaml放到某个web服务器上，然后通过wget去下载到core用户的当前目录下，如：

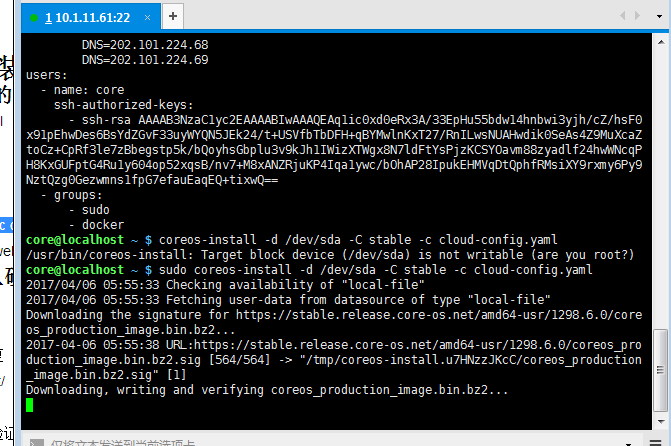
1. wget http://192.168.1.104/cloud-config.yaml

### 5、安装

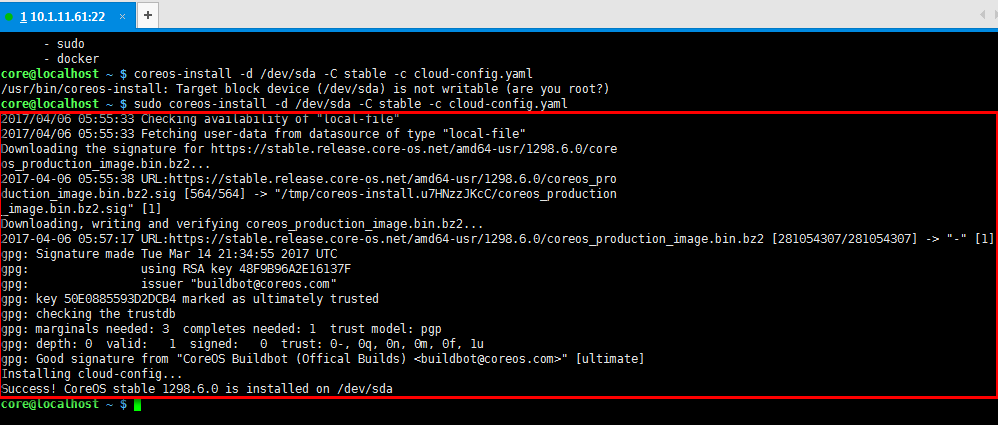
安装的命令为：coreos-install

|  |
| --- |
| USAGE="Usage: $0 [-V version] [-d /dev/device]  Options:  -d DEVICE Install CoreOS to the given device.  -V VERSION Version to install (e.g. current) [default: ${VERSION\_ID}]  -B BOARD CoreOS board to use [default: ${BOARD}]  -C CHANNEL Release channel to use (e.g. beta) [default: ${CHANNEL\_ID}]  -o OEM OEM type to install (e.g. ami) [default: ${OEM\_ID:-(none)}]  -c CLOUD Insert a cloud-init config to be executed on boot.  -i IGNITION Insert an Ignition config to be executed on boot.  -t TMPDIR Temporary location with enough space to download images.  -v Super verbose, for debugging.  -b BASEURL URL to the image mirror (overrides BOARD)  -n Copy generated network units to the root partition.  -h This ;-)  This tool installs CoreOS on a block device. If you PXE booted CoreOS on a  machine then use this tool to make a permanent install. |

1、在ssh中 输入：coreos-install -d /dev/sda -C stable -c cloud-config.yaml，即可安装，



上面中首先从外网中下载安装的文件和验证文件，没墙的话，大约二十分钟左右即可安装完成



【可选方案】

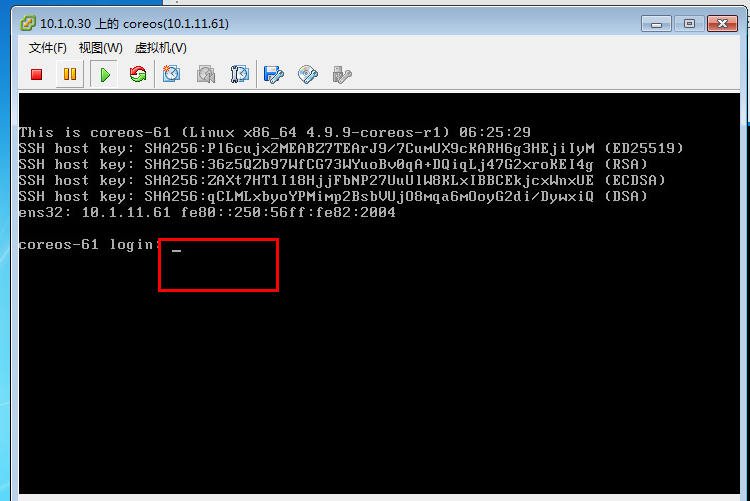
首先搭建web服务器，把上图中下载的文件两个文件预先下载下来，放到web服务器中current文件夹中（没有创建）。然后在ssh中运行

coreos-install -d /dev/sda -C stable -c cloud-config.yaml -b <http://10.1.11.20>:80 -v current

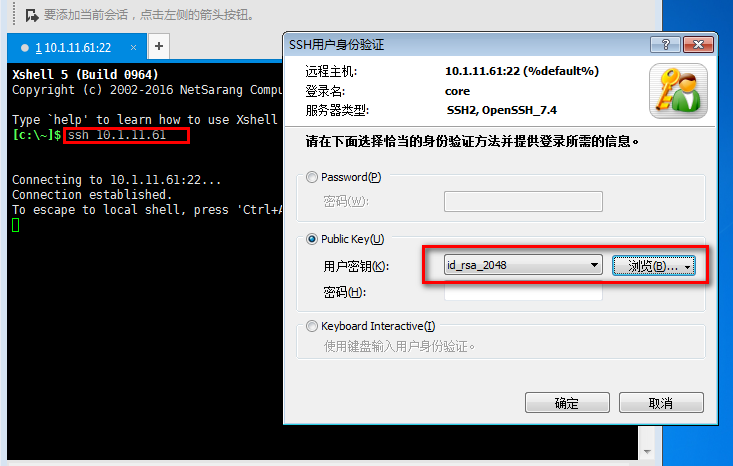
即可以从内网中的web服务器中下载和安装。

### 6、安装之后登录操作

现在CoreOS可以正常启动了，但是登录不进去

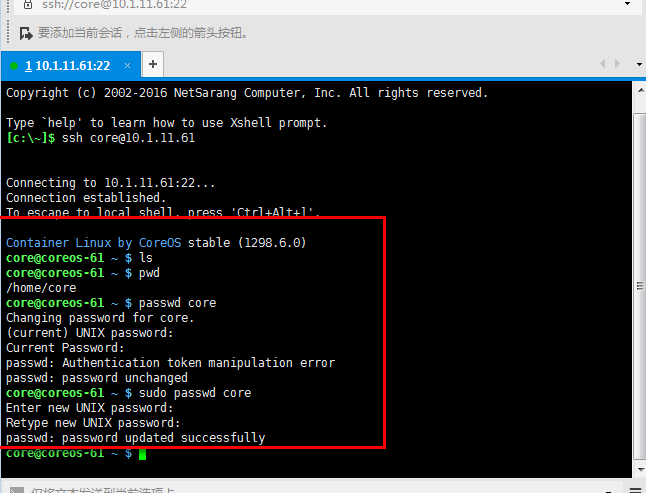


1、在ssh客户端（xshell）,先ssh [core@10.1.11.61](mailto:core@10.1.11.61),然后在弹出的对话中选择第四步中保存的私钥。



如何制定公钥时指定密码，上图中也需要指定密码。

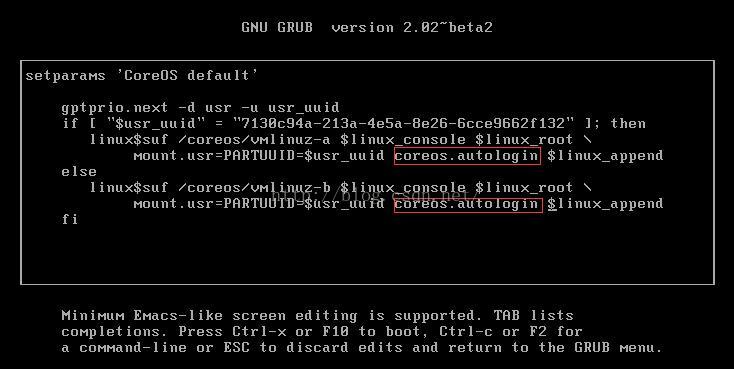
2、通过ssh进入coreos之后，修改core的密码。



3、接下来在coreos的虚拟机就可以按新的密码登录了。

【可选方案】

. 现在CoreOS可以正常启动了，但是登录不进去，重启CoreOS，在界面出现bootloader时，按下 e 键，进入GRUB 菜单，进行编辑，添加红色框里的内容 coreos.autologin。修改完成后 按F10，重新启动，就可以免密码登录进入了，默认用户是core（注意：很难进入到下面的界面，按e键要非常快）



console=tty0 console=ttyS0 coreos.autologin=tty1 coreos.autologin=ttyS0

进入之后 sudo su 进入超级权限，并修改用户core密码

## 附一 parted创建GPT分区（fdisk不支持创建GPT分区，GPT支持大于2TB分区，MBR不支持）

 这次干活，服务器硬盘26T，但是发现无论是Ubuntu11.04还是Centos5.6，都是无法直接分配一个分区超过16T.

没想到，**[Linux](http://lib.csdn.net/base/linux" \t "_blank" \o "Linux知识库)**还有这个局限。改天好好研究一下这个。

2011年03月19日 星期六 17:18

本次操作环境：

Ubuntu Server 10.10

SCSI Harddisk：/dev/sda       500GB

U盘：/dev/sdb    8GB（模拟成USB Harddisk，安装OS）

介绍2种分区表：  
MBR分区表：（MBR含义：主引导记录）  
所支持的最大卷：2T （T; terabytes,1TB=1024GB）  
对分区的设限：最多4个主分区或3个主分区加一个扩展分区。  
  
GPT分区表：（GPT含义：GUID分区表）  
支持最大卷：18EB，（E：exabytes,1EB=1024TB）  
每个磁盘最多支持128个分区

所以如果要大于2TB的卷或分区就必须得用GPT分区表。

Linux下fdisk工具不支持GPT，得使用另一个GNU发布的强大分区工具parted。

fdisk工具用的话，会有下面的警告信息：

**WARNING: GPT (GUID Partition Table) detected on '/dev/sda'! The util fdisk doesn't support GPT. Use GNU Parted.**

**下面是用parted工具对/dev/sda做GPT分区的过程：**

root@node01:/mnt# **parted /dev/sda**  
GNU Parted 2.3  
Using /dev/sda  
Welcome to GNU Parted! Type 'help' to view a list of commands.

(parted) **mklabel gpt**                                                        
Warning: The existing disk label on /dev/sda will be destroyed and all data on this disk  
will be lost. Do you want to continue?  
Yes/No? **yes**

(parted) **print**                                                             
Model: DELL PERC 6/i Adapter (scsi)  
Disk /dev/sda: 500GB  
Sector size (logical/physical): 512B/512B  
Partition Table: gpt  
  
Number  Start  End  Size  File system  Name  Flags

(parted)**mkpart primary 0KB 500GB**  
Warning: You requested a partition from 0.00B to 500GB.                     
The closest location we can manage is 17.4kB to 500GB.  
Is this still acceptable to you?  
Yes/No? **yes**                                                                 
Warning: The resulting partition is not properly aligned for best performance.  
Ignore/Cancel? **Ignore**

(parted) **print**                                                             
Model: DELL PERC 6/i Adapter (scsi)  
Disk /dev/sda: 500GB  
Sector size (logical/physical): 512B/512B  
Partition Table: gpt  
  
Number  Start   End    Size   File system  Name     Flags  
 1      17.4kB  500GB  500GB               primary

(parted)**quit**                                                              
Information: You may need to update /etc/fstab.                             
  
root@node01:/#**fdisk -l**  
  
WARNING: GPT (GUID Partition Table) detected on '/dev/sda'! The util fdisk doesn't support GPT. Use GNU Parted.  
  
  
Disk /dev/sda: 499.6 GB, 499558383616 bytes  
255 heads, 63 sectors/track, 60734 cylinders  
Units = cylinders of 16065 \* 512 = 8225280 bytes  
Sector size (logical/physical): 512 bytes / 512 bytes  
I/O size (minimum/optimal): 512 bytes / 512 bytes  
Disk identifier: 0x00000000  
  
   Device Boot      Start         End      Blocks   Id  System  
/dev/sda1               1       60735   487849983+  ee  GPT  
  
root@node01:/#**mkfs.ext4 /dev/sda1**  
mke2fs 1.41.12 (17-May-2010)  
文件系统标签=  
[**操作系统**](http://lib.csdn.net/base/operatingsystem):Linux  
块大小=4096 (log=2)  
分块大小=4096 (log=2)  
Stride=0 blocks, Stripe width=0 blocks  
30490624 inodes, 121962487 blocks  
6098124 blocks (5.00%) reserved for the super user  
第一个数据块=0  
Maximum filesystem blocks=4294967296  
3722 block groups  
32768 blocks per group, 32768 fragments per group  
8192 inodes per group  
Superblock backups stored on blocks:   
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,   
        4096000, 7962624, 11239424, 20480000, 23887872, 71663616, 78675968,   
        102400000  
  
正在写入inode表: 完成                              
Creating journal (32768 blocks): 完成  
Writing superblocks and filesystem accounting information: 完成  
  
This filesystem will be automatically checked every 24 mounts or  
180 days, whichever comes first.  Use tune2fs -c or -i to override.

root@node01:/#**mount /dev/sda1 /export/glusterfs01/**

root@node01:/# **df -h**  
Filesystem            Size  Used Avail Use% Mounted on  
/dev/mapper/node01-root  
                      6.8G  987M  5.5G  16% /  
none                  7.9G  208K  7.9G   1% /dev  
none                  7.9G     0  7.9G   0% /dev/shm  
none                  7.9G   32K  7.9G   1% /var/run  
none                  7.9G     0  7.9G   0% /var/lock  
/dev/sdb1             228M   21M  196M  10% /boot  
/dev/sda1             458G  198M  435G   1% /export/glusterfs01

root@node01:/#**vi /etc/fstab**  
  
# /etc/fstab: static file system information.  
#  
# Use 'blkid -o value -s UUID' to print the universally unique identifier  
# for a device; this may be used with UUID= as a more robust way to name  
# devices that works even if disks are added and removed. See fstab(5).  
#  
# <file system> <mount point>   <type>  <options>       <dump>  <pass>  
proc            /proc           proc    nodev,noexec,nosuid 0       0  
/dev/mapper/node01-root /               ext4    errors=remount-ro 0       1  
# /boot was on /dev/sdb1 during installation  
UUID=c21707ff-ba0f-43ee-819a-8e72fa0f8500 /boot           ext2    defaults        0       2  
/dev/mapper/node01-swap\_1 none            swap    sw              0       0  
**/dev/sda1       /export/glusterfs01     ext4    defaults        0       2**

重启就可以自动挂载了！至此完成。

## 集群安装

|  |
| --- |
| etcd2:  *# generate a new token for each unique cluster from https://discovery.etcd.io/new?size=3*  discovery: "https://discovery.etcd.io/<token>"  *# multi-region and multi-cloud deployments need to use $public\_ipv4*  advertise-client-urls: "http://$public\_ipv4:2379"  initial-advertise-peer-urls: "http://$private\_ipv4:2380"  *# listen on both the official ports and the legacy ports*  *# legacy ports can be omitted if your application doesn't depend on them*  listen-client-urls: "http://0.0.0.0:2379,http://0.0.0.0:4001"  listen-peer-urls: "http://$private\_ipv4:2380,http://$private\_ipv4:7001" |

<https://discovery.etcd.io/new?size=3> 得到的

https://discovery.etcd.io/db590f81044454e618f635e860687c5f

https://coreos.com/os/docs/1367.5.0/cloud-config.html

|  |
| --- |
| #cloud-config  hostname: coreos-61  coreos:  etcd2:  discovery: " https://discovery.etcd.io/db590f81044454e618f635e860687c5f "  advertise-client-urls: "http://$public\_ipv4:2379"  initial-advertise-peer-urls: "http://$private\_ipv4:2380"  listen-client-urls: "http://0.0.0.0:2379,http://0.0.0.0:4001"  listen-peer-urls: <http://$private_ipv4:2380,http://$private_ipv4:7001>  update:  reboot-strategy: "off"  units:  - name: etcd2.service  command: start  - name: fleet.service  command: start  - name: flanneld.service  drop-ins:  - name: 50-network-config.conf  Content: |  [service]  ExecStartPre=/usr/bin/etcdctl set /coreos.com/network/config '{ "Network" : "10.1.0.0/16" }'  command: start  - name: static.network  content: |  [Match]  Name=ens32  [Network]  Address=10.1.11.80/24  Gateway=10.1.11.254  DNS=202.101.224.68  DNS=202.101.224.69  users:  - name: core  ssh-authorized-keys:  - ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAq1ic0xd0eRx3A/33EpHu55bdw14hnbwi3yjh/cZ/hsF0x91pEhwDes6BsYdZGvF33uyWYQN5JEk24/t+USVfbTbDFH+qBYMwlnKxT27/RnILwsNUAHwdik0SeAs4Z9MuXcaZtoCz+CpRf3le7zBbegstp5k/bQoyhsGbplu3v9kJh1IWizXTWgx8N7ldFtYsPjzKCSYOavm88zyadlf24hwWNcqPH8KxGUFptG4Ru1y604op52xqsB/nv7+M8xANZRjuKP4Iqa1ywc/bOhAP28IpukEHMVqDtQphfRMsiXY9rxmy6Py9NztQzg0Gezwmns1fpG7efauEaqEQ+tixwQ==  - groups:  - sudo  - docker |

|  |
| --- |
| #cloud-config  hostname: coreos-61  coreos:  etcd:  addr: $private\_ipv4:4001  peer-addr: $private\_ipv4:7001  units:  - name: etcd.service  command: start  - name: fleet.service  command: start  - name: static.network  content: |  [Match]  Name=ens32  [Network]  Address=10.1.11.61/24  Gateway=10.1.11.254  DNS=202.101.224.68  DNS=202.101.224.69  users:  - name: core  ssh-authorized-keys:  - ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAq1ic0xd0eRx3A/33EpHu55bdw14hnbwi3yjh/cZ/hsF0x91pEhwDes6BsYdZGvF33uyWYQN5JEk24/t+USVfbTbDFH+qBYMwlnKxT27/RnILwsNUAHwdik0SeAs4Z9MuXcaZtoCz+CpRf3le7zBbegstp5k/bQoyhsGbplu3v9kJh1IWizXTWgx8N7ldFtYsPjzKCSYOavm88zyadlf24hwWNcqPH8KxGUFptG4Ru1y604op52xqsB/nv7+M8xANZRjuKP4Iqa1ywc/bOhAP28IpukEHMVqDtQphfRMsiXY9rxmy6Py9NztQzg0Gezwmns1fpG7efauEaqEQ+tixwQ==  - groups:  - sudo  - docker |