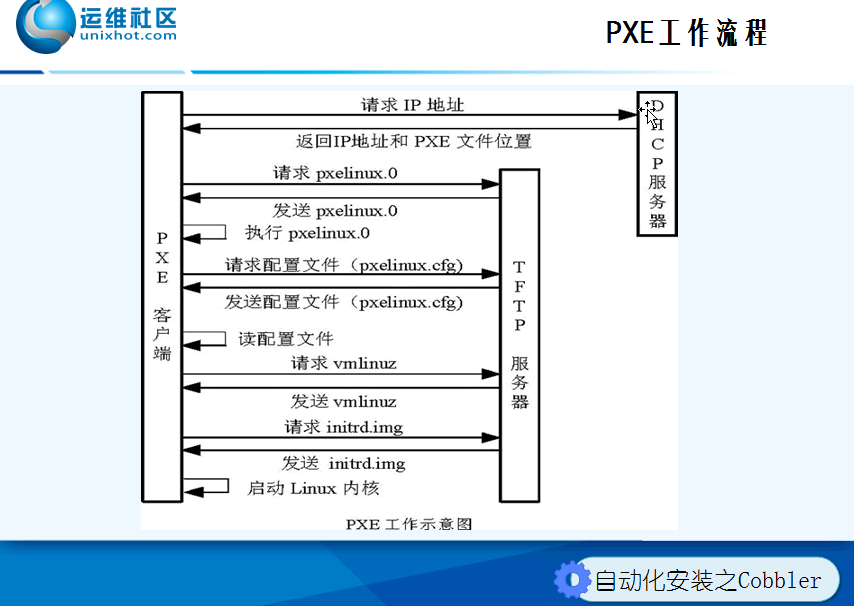
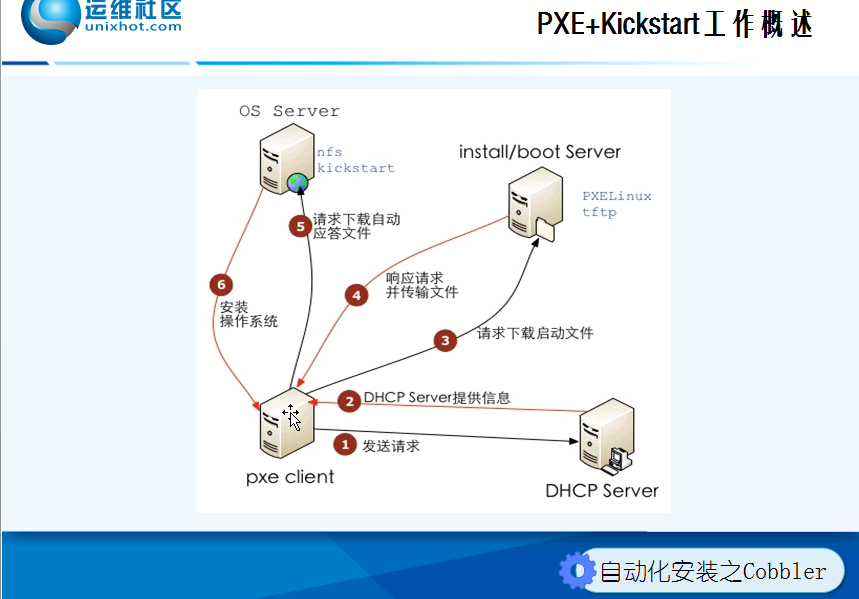
# 自动化运维至自动化安装Cobbler

## 1、PXT工作流程：



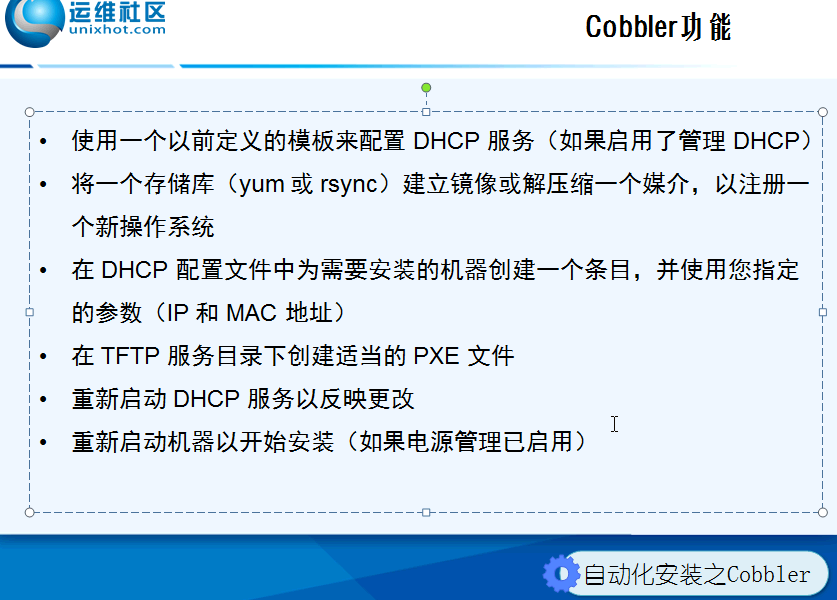
1. PXE客户端启动时选择或设置为**网卡启动，**向本网络中的DHCP服务器申请IP信息，包括IP地址、掩码、网关等
2. DHCP服务器返回给客户机一个IP地址和PXE文件的位置（该文件一般是放在一台TFTP服务器上）。
3. PXE客户端向本网络中的TFTP服务器获取pxelinux.0文件
4. PXE客户端取得pxelinux.0后执行该文件，并且向TFTP服务器请求获取pxelinux.cfg配置文件
5. PXE客户端通过读取配置文件，请求获得vmlinuz和initrd.img,最后启动linux内核
6. 进入安装界面通过选择HTTP,FTP,NFS方式之一进行安装。

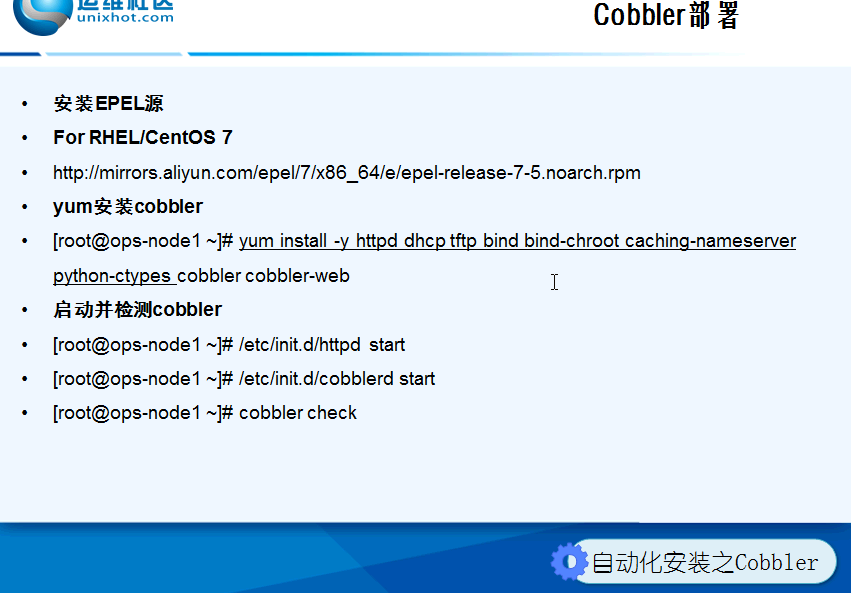
## 2、PXE+Kickstart工作概述



* PXE+Kickstart工作概述
* 配置服务，比如DHCP,TFTP,(HTTP,FTP和NFS)
* 创建自动部署文件（比如说keckstart）
* 讲安装媒介解压缩到HTTP/FTP/NFS存储库中

## 3、Cobbler补鞋匠安装部署





### （1）安装Cobbler及相关软件

yum install -y httpd dhcp tftp python-ctypes cobbler cobbler-web pykickstart xinetd

其中：

* httpd和cobbler-web 提供cobbler的web图形管理界面
* dhcp提供IP地址分配服务
* xinetd是一个超级守护进程，管理无法开机启动的服务如tftp开机启动

### （2）Cobbler的主配置文件目录

以上安装包的cobbler是最核心的安装包，其安装后的主要目录如下：

/etc/cobbler

#它包含了cobbler的默认配置文件：/etc/cobbler/settings

#DHCP服务的配置模板：/etc/cobbler/dhcp.template 打开DHCP管理后可以根据此文件生成dhcpd.conf文件

#tftp服务的配置模板：/etc/cobbler/tftpd.template

#rsync服务的配置模板：/etc/cobbler/rsync.template

#iso模板配置文件目录：/etc/cobbler/iso

#pxe模板文件目录：/etc/cobbler/pxe

#电源配置文件目录：/etc/cobbler/power

#web服务授权配置文件：/etc/cobbler/users.conf

#web访问的用户名密码配置文件：/etc/cobbler/users.digest

#DNS服务的配置模板：/etc/cobbler/dnsmasq.template

#Cobbler模块配置文件：/etc/cobbler/modules.conf

/var/lib/cobbler

# Cobbler数据目录，它包含了cobbler数据配置文件：/var/lib/cobbler/config

# 默认存放kickstart文件：/var/lib/cobbler/kickstarts

# 存放的各种引导程序：/var/lib/cobbler/loaders

# 系统安装镜像目录：/var/www/cobbler

# 导入的系统镜像列表：/var/www/cobbler/ks\_mirror

# 导入的系统镜像启动文件：/var/www/cobbler/images

# yum源存储目录：/var/www/cobbler/repo\_mirror

# 日志目录：/var/log/cobbler

# 客户端系统安装日志：/var/log/cobbler/install.log

# cobbler日志：/var/log/cobbler/cobbler.log

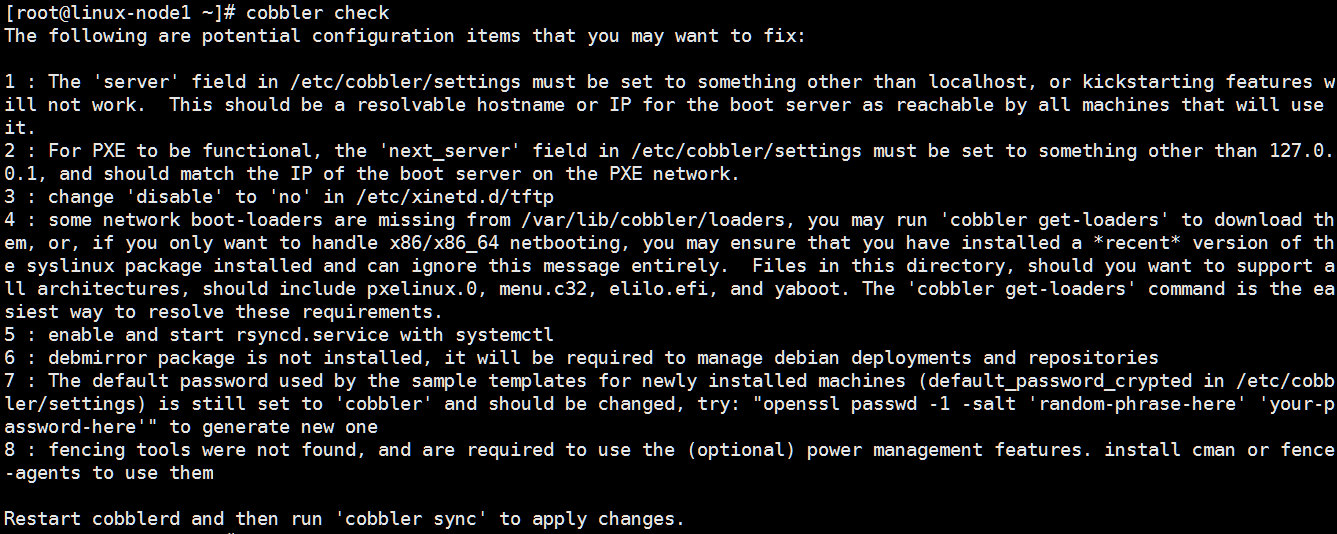
### （3）启动cobbler主服务和httpd

systemctl start cobblerd

systemctl start httpd

### （4）检查cobbler环境命令cobbler check

cobbler check环境检查会有8个报错分别是：



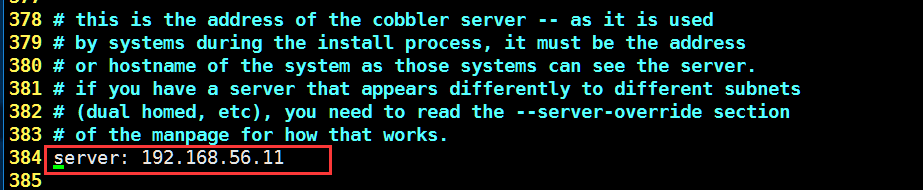
#### 第一个错误：

1 : The 'server' field in /etc/cobbler/settings must be set to something other than localhost, or kickstarting features will not work. This should be a resolvable hostname or IP for the boot server as reachable by all machines that will use it.

**解决办法**：

sed -i ‘s#server: 127.0.0.1#server:192.168.56.11#g’ /etc/cobbler/settings #改为本机的IP地址

或通过vim 打开/etc/cobbler/settings进行修改



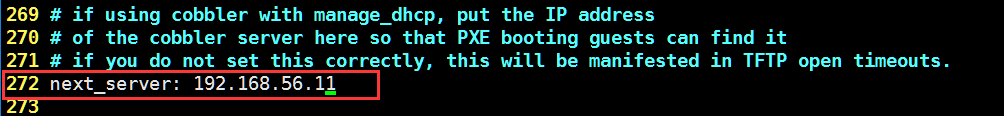
#### 第二个错误

2 : For PXE to be functional, the 'next\_server' field in /etc/cobbler/settings must be set to something other than 127.0.0.1, and should match the IP of the boot server on the PXE network.

**解决办法**:

sed -i ‘s#next\_server:127.0.0.1#next\_server: 192.168.56.11#g’ /etc/cobbler/settings #改为本机的IP地址

或通过vim 更改为本机IP地址



#### 第三个错误

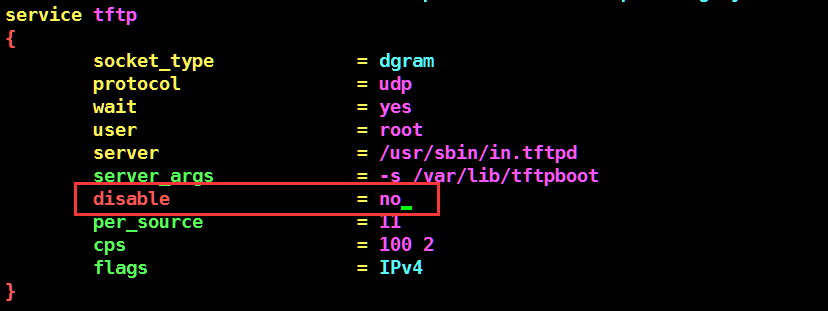
3 : change 'disable' to 'no' in /etc/xinetd.d/tftp

**解决办法**：

vim /etc/xinetd.d/tftp

将disable = yes 改为 disable = no

systemctl restart xinetd

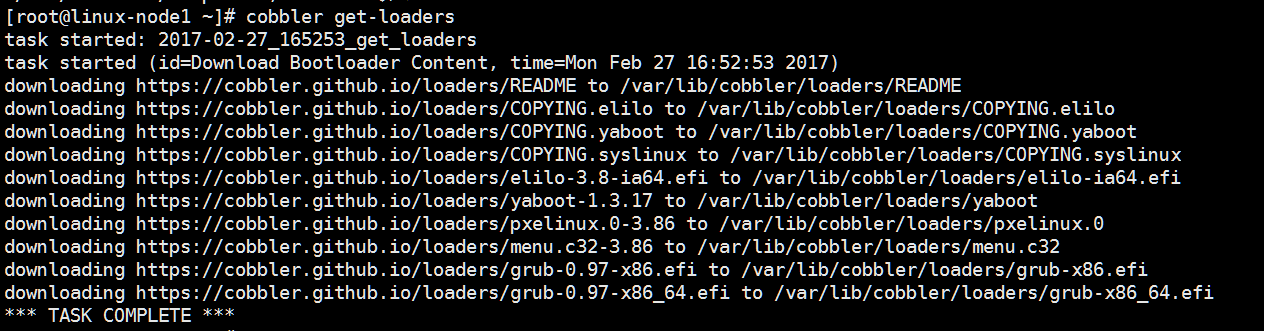


#### 第四个错误

4 : some network boot-loaders are missing from /var/lib/cobbler/loaders, you may run 'cobbler get-loaders' to download them, or, if you only want to handle x86/x86\_64 netbooting, you may ensure that you have installed a \*recent\* version of the syslinux package installed and can ignore this message entirely. Files in this directory, should you want to support all architectures, should include pxelinux.0, menu.c32, elilo.efi, and yaboot. The 'cobbler get-loaders' command is the easiest way to resolve these requirements.

**解决方法**：

执行**cobbler get-loaders** #会下载相关的启动文件



#### 第五个错误

5 : enable and start rsyncd.service with systemctl

**解决办法**:

systemctl enable rsyncd

systemctl start rsyncd

#### 第六个错误

6 : debmirror package is not installed, it will be required to manage debian deployments and repositories

**这个报错可以忽略，如果你是强迫症的话可以执行以下操作：**

yum -y install debmirror

注释/etc/debmirror.conf文件中的 #dists="sid"; @arches="i386"

#### 第七个错误

7 : The default password used by the sample templates for newly installed machines (default\_password\_crypted in /etc/cobbler/settings) is still set to 'cobbler' and should be changed, try: "openssl passwd -1 -salt 'random-phrase-here' 'your-password-here'" to generate new one

**解决办法**：

openssl passwd -1 -salt "随机短语" "密码"

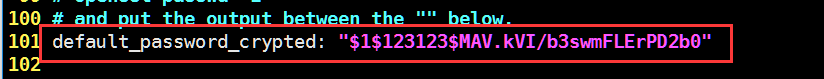
openssl passwd -1 -salt "123123" "123123"

$1$123123$MAV.kVI/b3swmFLErPD2b0

将生成的密码复制到 /etc/cobbler/settings

在 /etc/cobbler/settings 里的第101行

default\_password\_crypted: $1$123123$MAV.kVI/b3swmFLErPD2b0



更改配置文件后，要重启服务：systemctl restart cobblerd

#### 第八个错误

8 : fencing tools were not found, and are required to use the (optional) power management features. install cman or fence-agents to use them

#fence设备，电源管理

yum -y install cman fence-agents

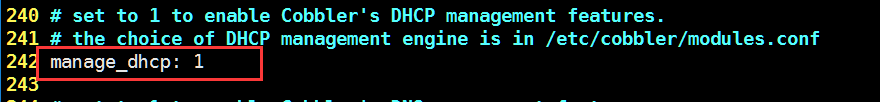
#### 最后重启cobbler

systemctl restart cobblerd

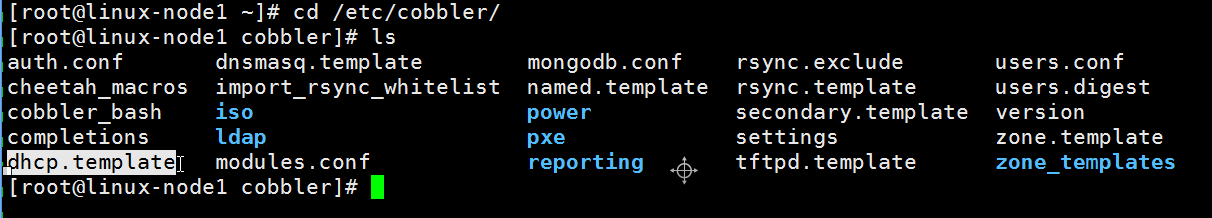
## 4、DHCP配置

### （1）配置通过cobbler管理dhcp

vim /etc/cobbler/settings #通过cobbler管理dhcp



### （2）修改模板文件



vim /etc/cobbler/dhcp.template

subnet 192.168.56.0 netmask 255.255.255.0 { #地址网段与掩码

option routers 192.168.56.2; #网关地址

option domain-name-servers 192.168.56.2; #DNS地址

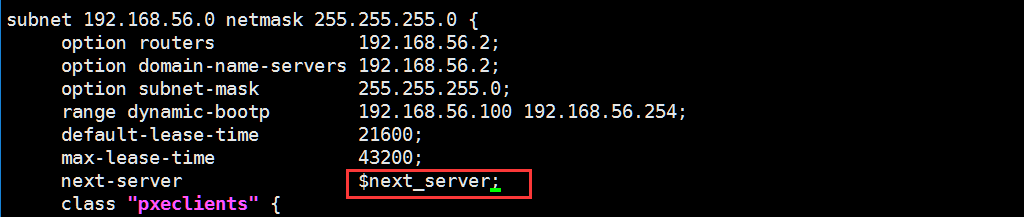
option subnet-mask 255.255.255.0; #子网掩码

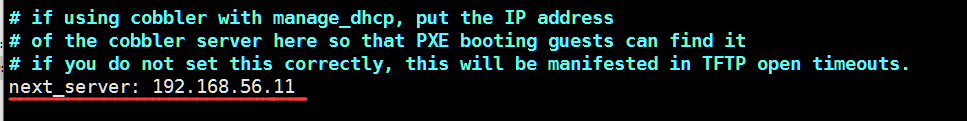
range dynamic-bootp 192.168.56.100 192.168.56.254; #分配的地址段

default-lease-time 21600;

max-lease-time 43200;

next-server $next\_server; #这里对应cobbler配置文件的变量





### （3）重启服务，执行cobbler sync生成dhcp模板

使用cobbler sync生成dhcp模板，自动帮我们重启dhcp

cobbler sync表示刷新，其实就是删除原来的文件，重新进行加载

systemctl restart cobblerd

cobbler sync

查看一下cobbler是否可以管理dhcp

[root@linux-node1 ~]# cat /etc/dhcp/dhcpd.conf

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Cobbler managed dhcpd.conf file

# generated from cobbler dhcp.conf template (Mon Feb 27 11:06:21 2017)

# Do NOT make changes to /etc/dhcpd.conf. Instead, make your changes

# in /etc/cobbler/dhcp.template, as /etc/dhcpd.conf will be

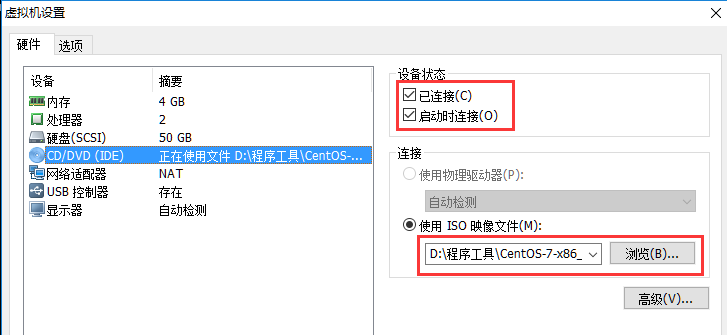
# overwritten.

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

上述意思是dhcp正在由cobbler进行管理，如果想修改，请修改/etc/cobbler/dhcp.template，然后/etc/dhcp/dhcpd.conf会被覆盖。

## 5、制作镜像

### （1）进行光盘挂载



[root@linux-node1 ~]# mount /dev/cdrom /mnt/

mount: /dev/sr0 is write-protected, mounting read-only

**把光盘挂载到/mnt/目录下**：

### （2）制作镜像

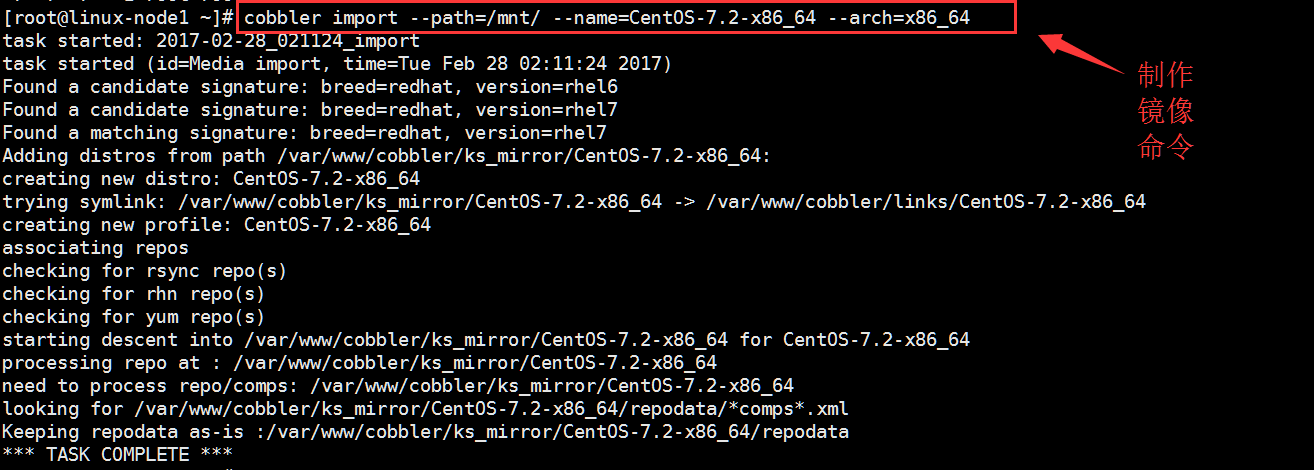
用于安装操作系统，是cobbler导入镜像

cobbler import --path=/mnt/ --name=CentOS-7.2-x86\_64 --arch=x86\_64

--path=导入路径

--name=镜像名称

--arch=系统位数



导入过程中centos7的本地yum源也就做好了，导入位置如下：

[root@linux-node1 ks\_mirror]# pwd

/var/www/cobbler/ks\_mirror

[root@linux-node1 ks\_mirror]# ls

CentOS-7.2-x86\_64 config

## 6、查看cobbler profile都有哪些命令

### （1）cobbler profile命令列表

[root@linux-node1 ks\_mirror]# cobbler profile

usage

=====

cobbler profile add

cobbler profile copy

cobbler profile dumpvars

cobbler profile edit

cobbler profile find

cobbler profile getks

cobbler profile list

cobbler profile remove

cobbler profile rename

cobbler profile report

### （2）查看当前服务器有几个镜像

[root@linux-node1 ~]# **cobbler profile list**

CentOS-7.2-x86\_64

### （3）查看系统镜像的详细信息

[root@linux-node1 ~]# **cobbler profile report**

Name : CentOS-7.2-x86\_64

TFTP Boot Files : {}

Comment :

DHCP Tag : default

Distribution : CentOS-7.2-x86\_64

Enable gPXE? : 0

Enable PXE Menu? : 1

Fetchable Files : {}

Kernel Options : {}

Kernel Options (Post Install) : {}

Kickstart : /var/lib/cobbler/kickstarts/sample\_end.ks

Kickstart Metadata : {}

Management Classes : []

Management Parameters : <<inherit>>

Name Servers : []

Name Servers Search Path : []

Owners : ['admin']

Parent Profile :

Internal proxy :

Red Hat Management Key : <<inherit>>

Red Hat Management Server : <<inherit>>

Repos : []

Server Override : <<inherit>>

Template Files : {}

Virt Auto Boot : 1

Virt Bridge : xenbr0

Virt CPUs : 1

Virt Disk Driver Type : raw

Virt File Size(GB) : 5

Virt Path :

Virt RAM (MB) : 512

Virt Type : kvm

## 7、修改kickstarts路径

当镜像系统自定义安装后，需要设置一些东西，例如哪些服务需要开启，哪些服务需要关闭，安装什么东西等。

### （1）我们先上传已经设置好的cfg文件

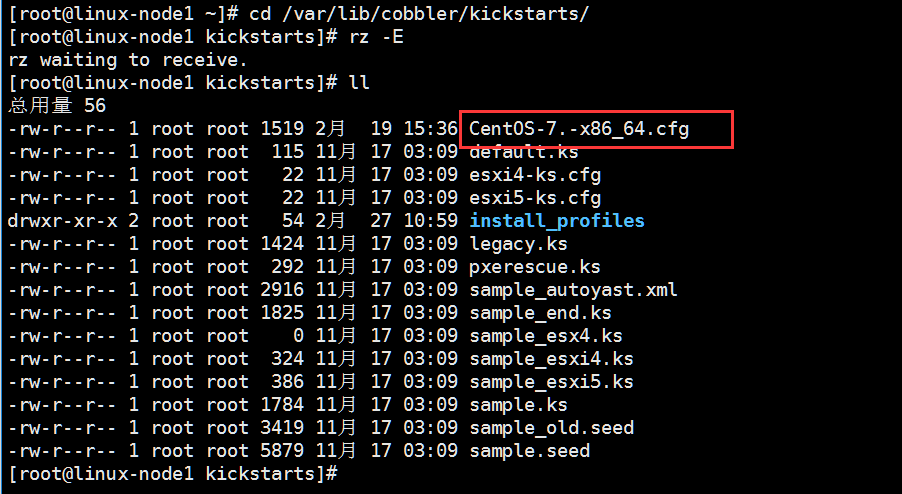
http://pan.baidu.com/s/1hsjWuX2#list/path=%2F

密码：iq4t

位了规范，我们把kickstart文件放在/var/lib/cobbler/kickstarts下面

检查文件路径是否正确

[root@linux-node1 ~]# cd /var/lib/cobbler/kickstarts/



### （2）自定义kickstarts文件

我们使用cobbler profile report命令

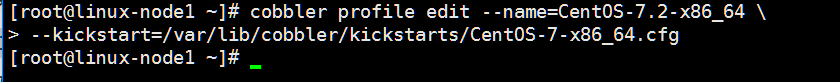
看到kickstart默认在/var/lib/cobbler/kickstarts/sample\_end.ks里

我们直接使用命令对其进行修改

修改CentOS7的kickstarts

cobbler profile edit --name=CentOS-7.2-x86\_64 \

--kickstart=/var/lib/cobbler/kickstarts/CentOS-7-x86\_64.cfg



注释：

--name=CentOS-7.2-x86\_64 是上传的镜像信息名，查询方法：**cobbler profile list**

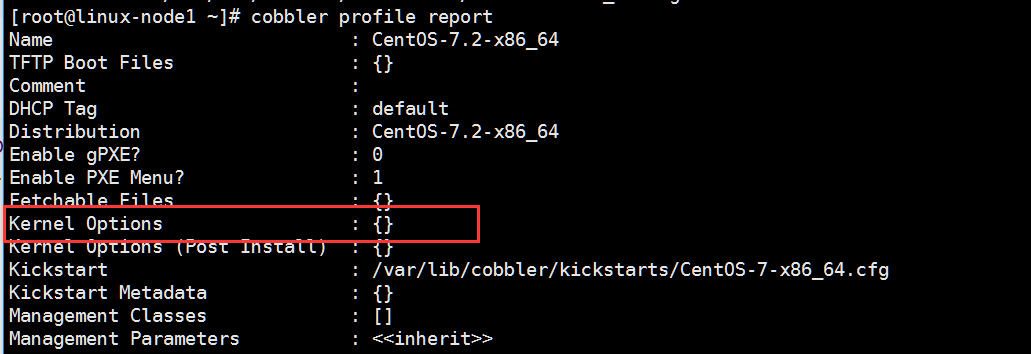
--kickstart=/var/lib/cobbler/kickstarts/CentOS-7-x86\_64.cfg 为kickstart储存路径

### （3）修改CentOS7默认网卡

由于CentOS7默认的网卡不在/etc/init.d/network

因此我们需要修改CentOS7的内核参数

我们用**cobbler profile report**查看系统版本内核信息如下

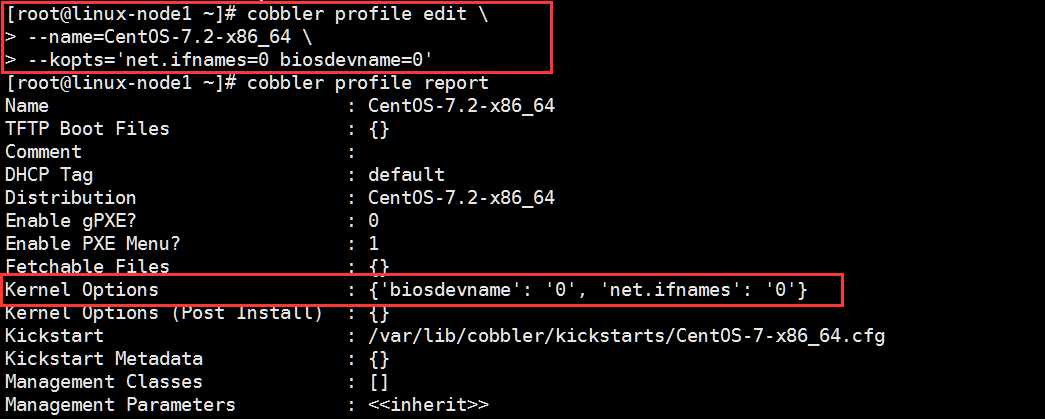


自定义内核参数

cobbler profile edit \

--name=CentOS-7.2-x86\_64 \

--kopts='net.ifnames=0 biosdevname=0'



进行上述修改后，CentOS7的网卡名称在自动安装完后就会变成eth0,eth1等等

温馨提示：默认 cobbler profile report 会查看当前配置得所有，我们可以使用 cobbler profile "Cobbler 定义的名字"来查看。

### （4）cobbler sync重新加载，使修改生效

执行cobbler sync会删除原来的文件，相当于重新进行加载，使修改过的配置文件生效

[root@linux-node1 ~]# cobbler sync

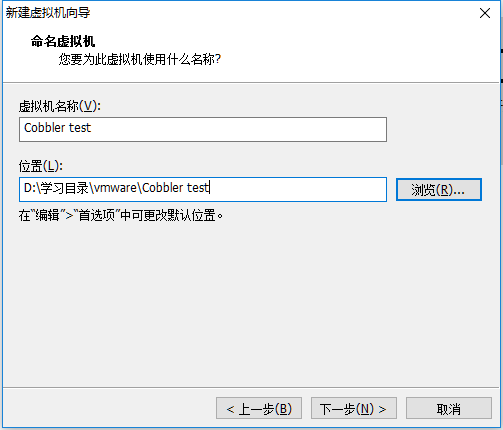
执行结果最后是：\*\*\* TASK COMPLETE \*\*\*

证明执行成功，太多了我就不截图了

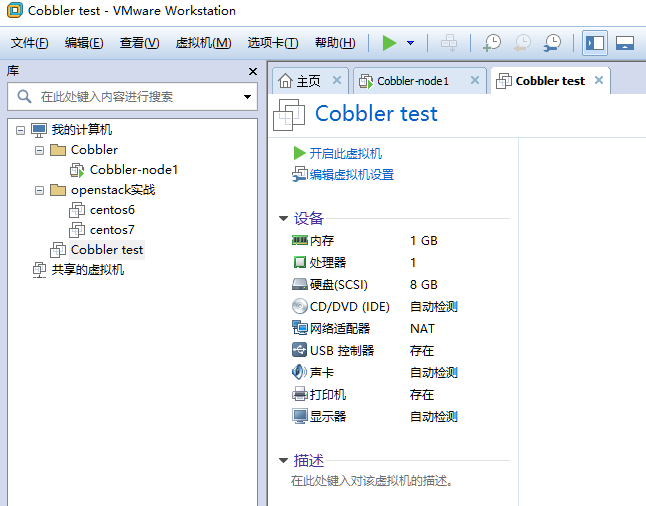
## 8、新建虚拟机进行安装

提示：我们先打开 系统日志，因为dhcp默认会将日志显示在/var/log/messages

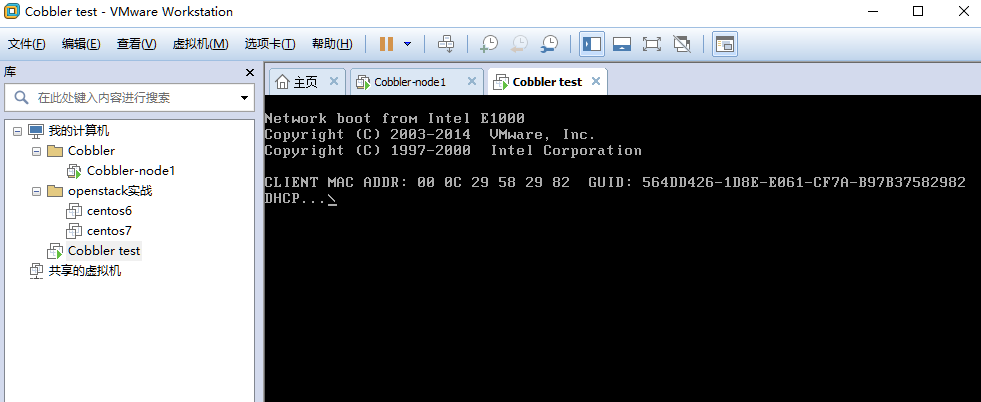
### 新建虚拟机

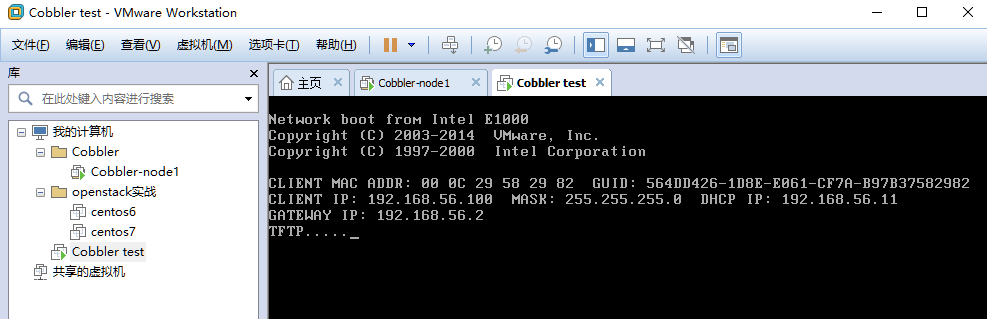


下一步，下一步，下一步，下一步。。。

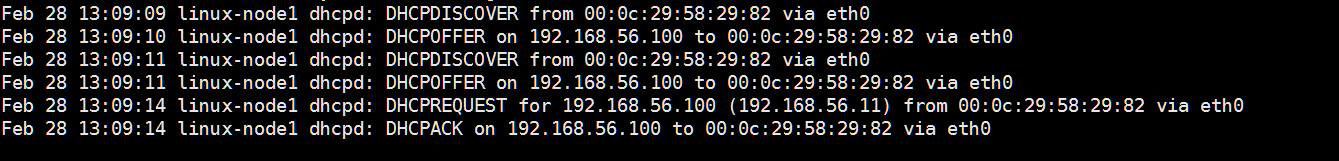


### （2）开启虚拟机

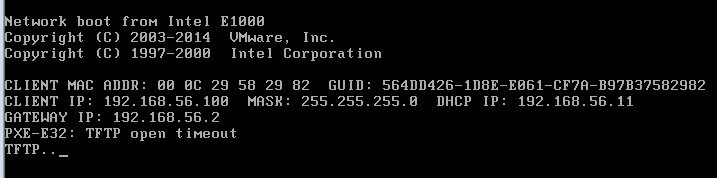




DHCP的4个步骤



现在已经获取到了IP，但是TFTP没有开启



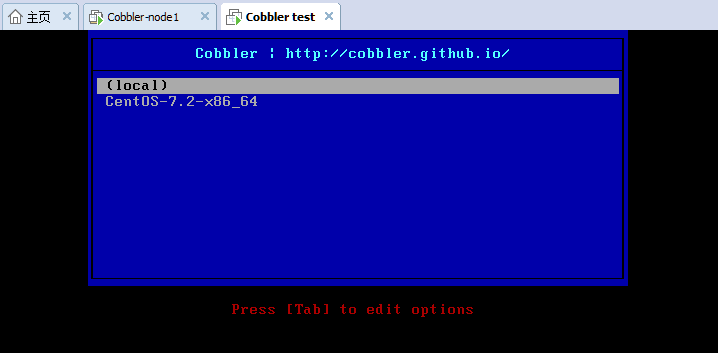
安装TFTP并且启动

[root@linux-node1 ~]# yum install xinetd -y

我们已经在开头的安装包组里安装过xinetd，直接启动即可

[root@linux-node1 ~]# systemctl start xinetd

### （3）重新启动虚拟机，进入系统选择界面



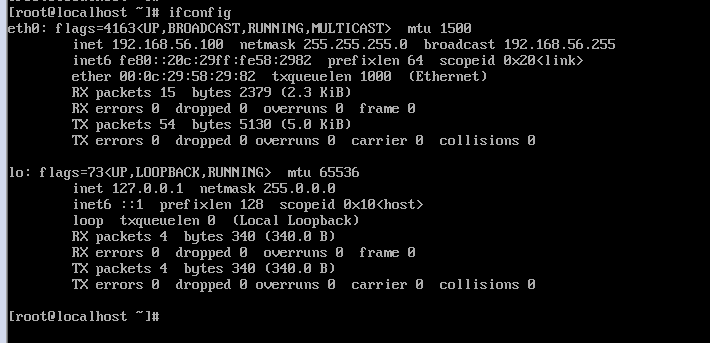
\*\* 菜单解释 \*\*

1. 本地硬盘
2. CentOS-7

提示：默认会进入倒计时，我们使用上下键取消倒计时。有这个菜单的好处就是不会担心重启之后就吧系统重装了。

这里我们选择安装 CentOS7

然后我们就可以坐下来喝杯茶了。



安装成功！

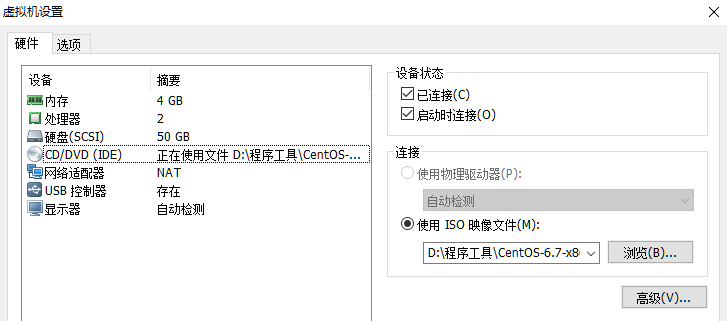
## 9、自动重装系统，使用koan工具

### （1）koan查看我们可以安装的系统

首先我们再挂载一个CentOS6的镜像->设置>CD/DVD

卸载CentOS7镜像，umount /dev/cdrom

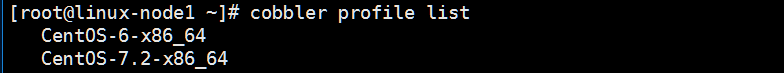
挂载CentOS6镜像，mount /dev/cdrom /mnt（挂载之前先按照下图加载镜像）



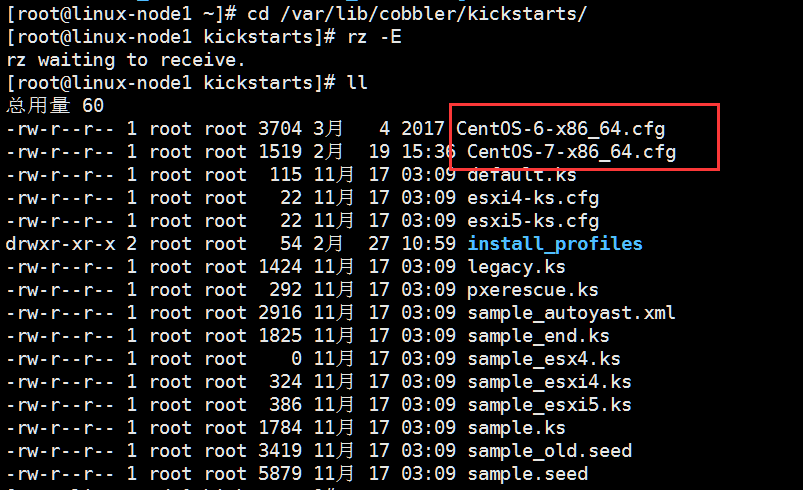
导入CentOS6镜像：

cobbler import --path=/mnt/ --name=CentOS-6-x86\_64 --arch=x86\_64





把设置好的cfg文件放到/var/lib/cobbler/kickstarts目录下面

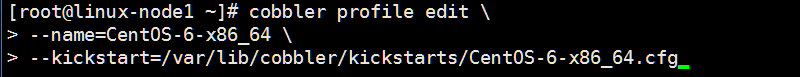


修改kickstarts文件路径

cobbler profile edit \

--name=CentOS-6-x86\_64 \

--kickstart=/var/lib/cobbler/kickstarts/CentOS-6-x86\_64.cfg



cobbler profile report查看修改结果



执行cobbler sync重新加载配置，使修改生效

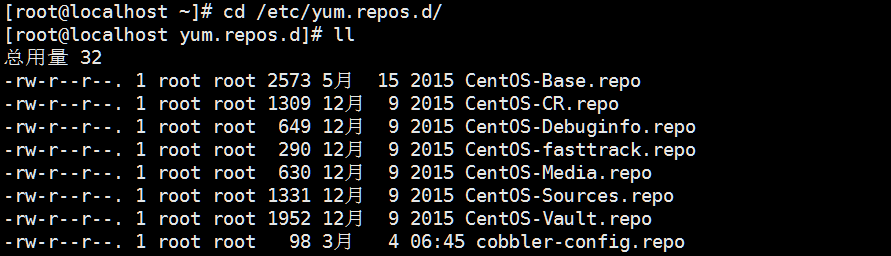
看到 \*\*\* TASK COMPLETE \*\*\* 就说明修改成功

### （2）安装koan

首先要在准备重装系统的服务器上，安装koan

[root@linux-node1 ~]# ssh 192.168.56.100 #刚才自动安装的那台服务器

给服务器安装repo源



wget http://mirrors.aliyun.com/repo/epel-7.repo

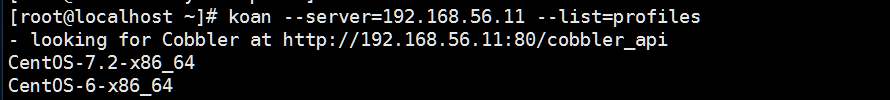
安装koan

[root@localhost ~]# yum install koan -y

### （3）使用koan查看cobbler服务器可以重装那些系统

使用koan命令查看可以安装的系统

[root@linux-node1 ~]# koan --server=192.168.56.11 --list=profiles

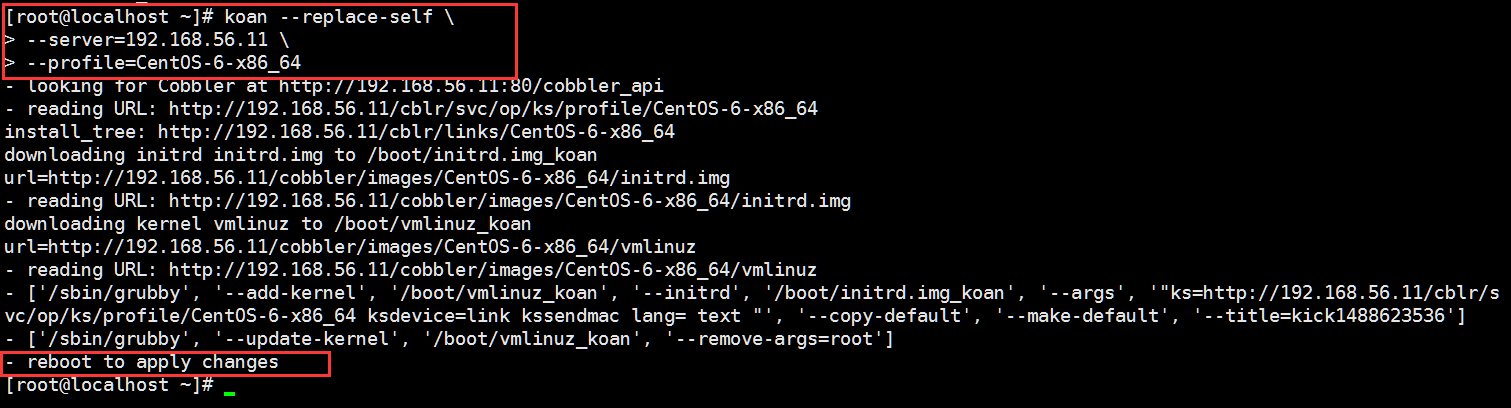


### （4）使用koan重装服务器

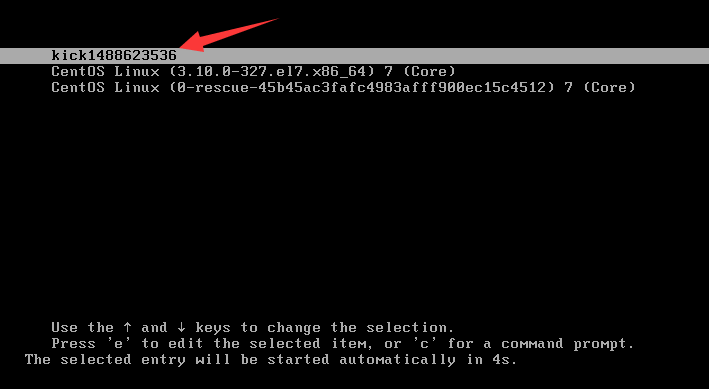
koan --replace-self \

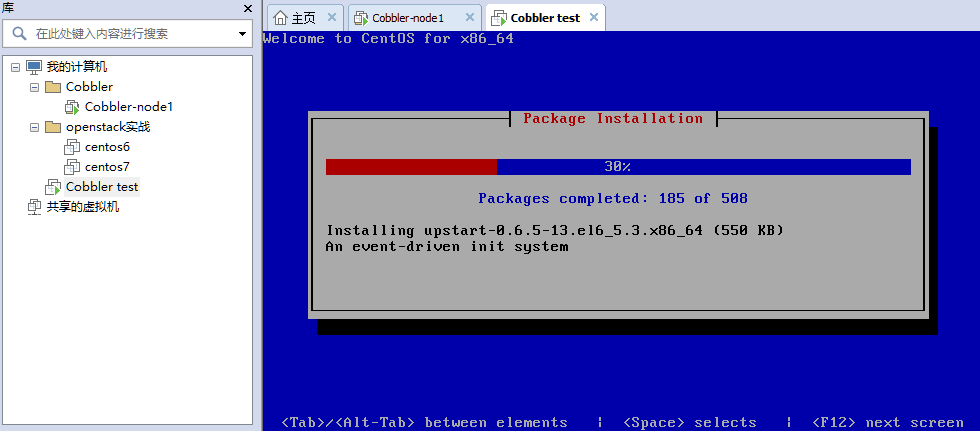
--server=192.168.56.11 \

--profile=CentOS-6-x86\_64

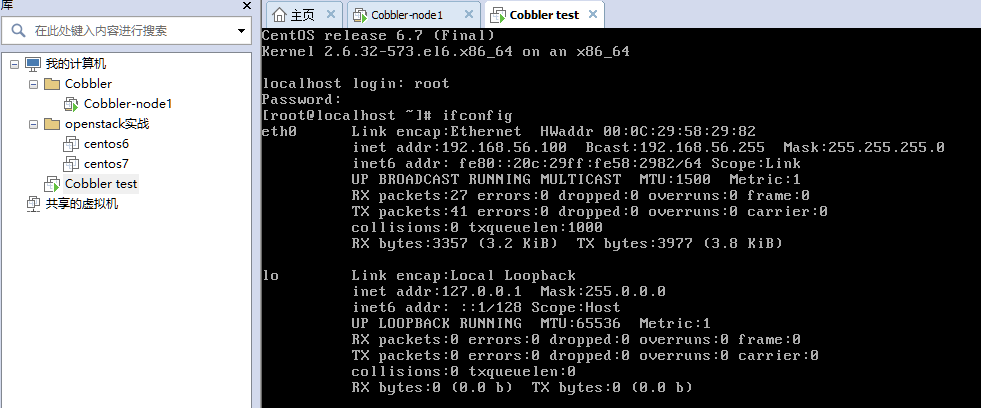


重启后发现，koan会在启动项加一条自动安装的选项，等待其自动执行即可



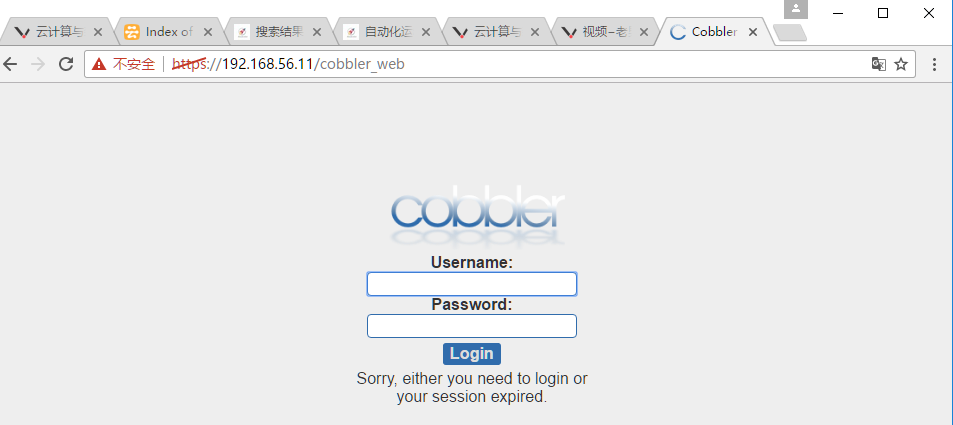


重装完毕



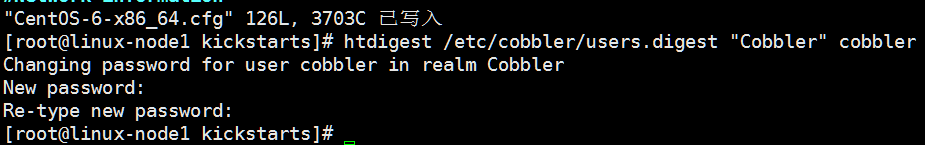
## 10、web页面登录cobbler

地址：https://192.168.56.11/cobbler\_web



（1）创建cobbler页面登录用户和密码

htdigest /etc/cobbler/users.digest "Cobbler" cobbler



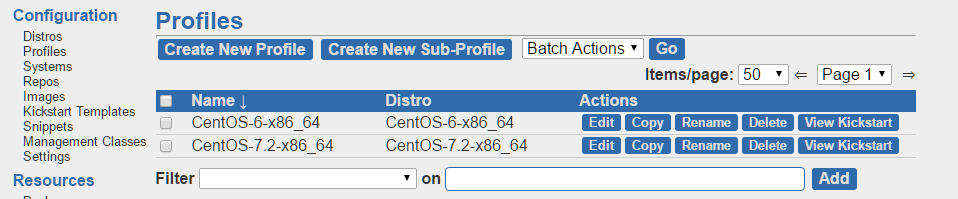
"Cobbler" 是用户的描述，随便写

cobbler 是用户名，也可以叫Jimmy

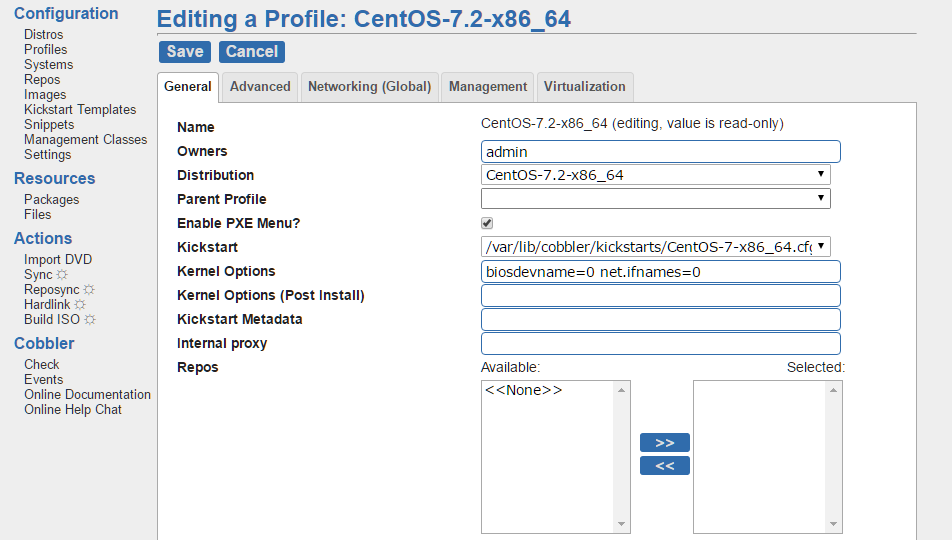


### （2）登录cobbler web

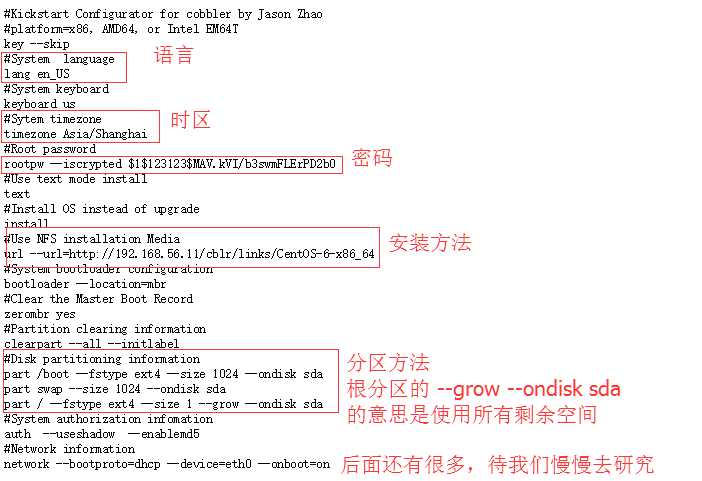
查看profiles，可以查看上传的镜像



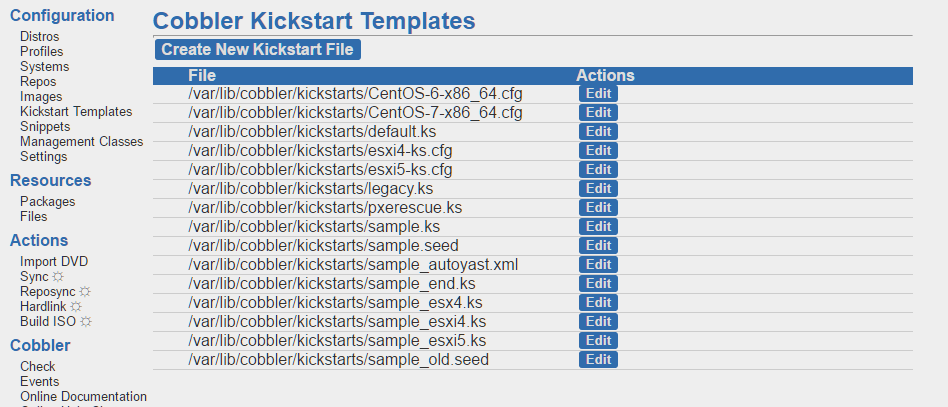
点击Edit可以修改系统的一些选项



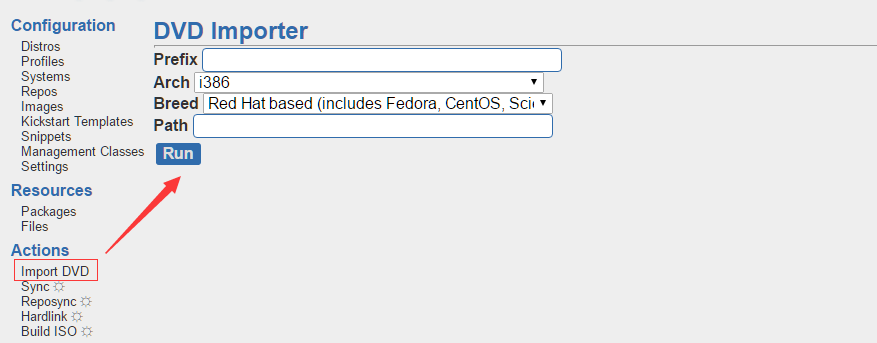
View Kickstart可以看到镜像的cfg文件



也可以在Kickstart Templates里面对系统内核进行修改

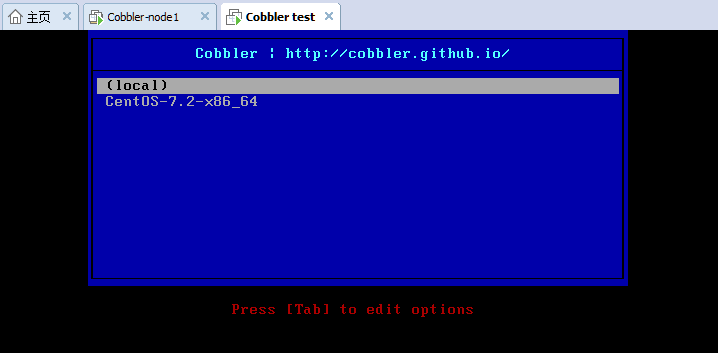


我们也可以修改kickstart



总之所有cobbler命令行操作，都可以在cobbler\_web上进行实现。

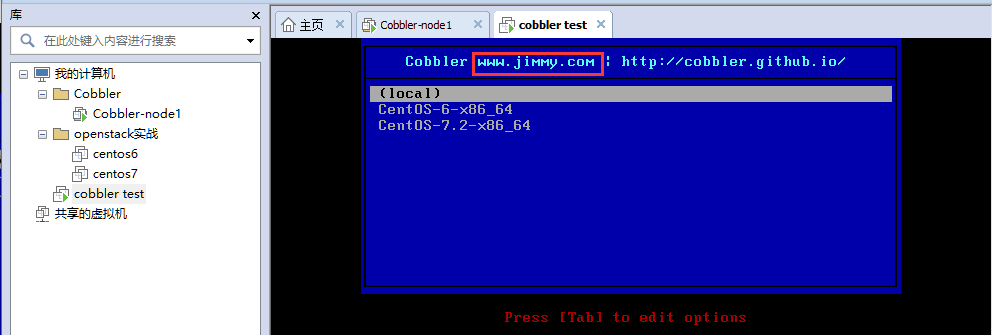
## 11、修改cobbler的选择系统菜单



编辑配置文件



重新创建虚拟机



## 12、Cobbler重要的几个目录

### （1）镜像存放目录

[root@linux-node1 ~]# ls /var/www/cobbler/ks\_mirror/

CentOS-6-x86\_64 CentOS-7.2-x86\_64 config

### （2）repo源仓库

[root@linux-node1 ~]# ls /var/www/cobbler/repo\_mirror/

### （3）kickstarts配置文件存放路径

[root@linux-node1 ~]# ls /var/lib/cobbler/kickstarts/

CentOS-6-x86\_64.cfg esxi4-ks.cfg legacy.ks sample\_end.ks sample\_esxi5.ks sample.seed

CentOS-7-x86\_64.cfg esxi5-ks.cfg pxerescue.ks sample\_esx4.ks sample.ks

default.ks install\_profiles sample\_autoyast.xml sample\_esxi4.ks sample\_old.seed

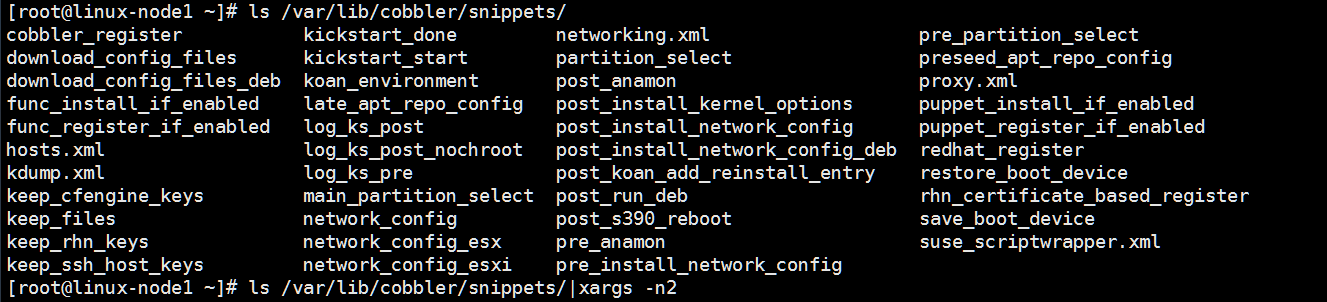
### （4）存放启动引导程序目录

[root@linux-node1 ~]# ls /var/lib/cobbler/loaders/

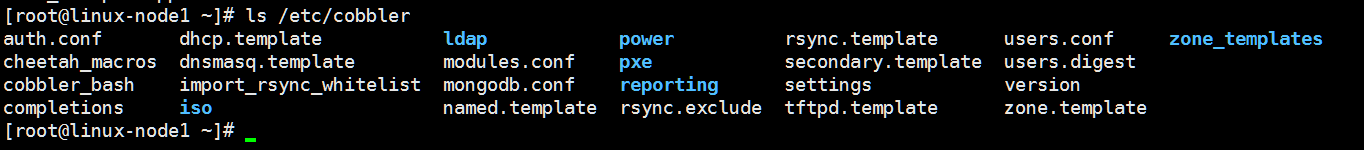
COPYING.elilo COPYING.syslinux COPYING.yaboot elilo-ia64.efi grub-x86\_64.efi

grub-x86.efi menu.c32 pxelinux.0 README yaboot

### （5）一些相关配置的脚本



### （6）cobbler配置文件存放目录



### （7）主要的配置文件目录还是settings

[root@linux-node1 ~]# ll /etc/cobbler/settings

-rw-r--r-- 1 root root 19654 2月 27 19:01 /etc/cobbler/settings

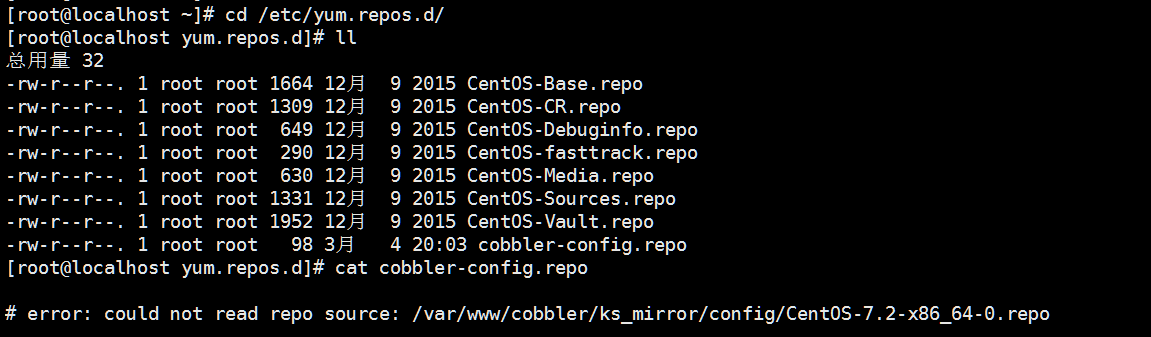
## 12B、构建yum仓库

当服务器装好后，我们还需要指定yum源才能对对所需配件进行安装

为了节省安装时间，我们可以直接在cobbler自动装机的时候进行yum源注入

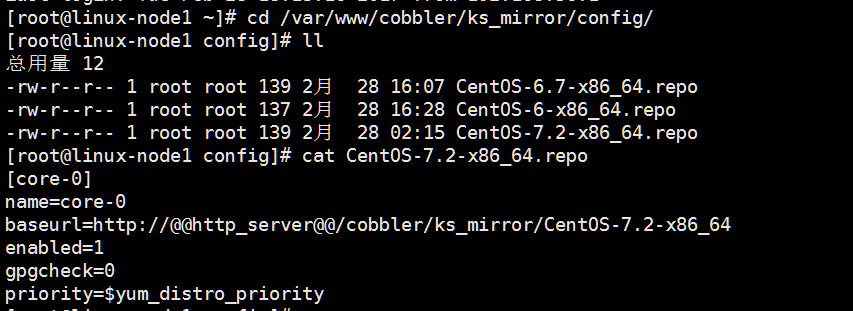
例如：我们构建一个openstack的yum源

### （1）给cobbler添加repo源



cobbler自动装机后，它会默认去找cobbler-config.repo里的repo文件

在cobbler服务器的对应目录/var/www/cobbler/ks\_mirror/config目录下实际上是有对应的repo文件的，但是路径不正确。



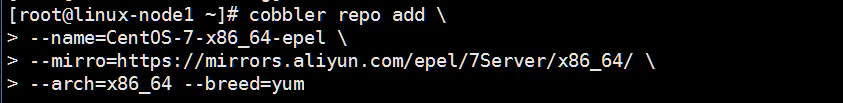
例如：我们现在需要导入其他的yum源，例如centos7的epel源

[root@linux-node1 ~]# cobbler repo add \

> --name=CentOS-7-x86\_64-epel \

> --mirro=https://mirrors.aliyun.com/epel/7Server/x86\_64/ \

> --arch=x86\_64 --breed=yum



**当心，如果是vmware虚拟机的实验环境，千万不要按回车，好几百个G的内容同步**

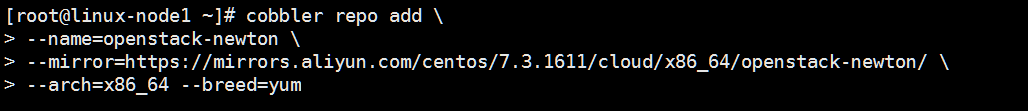
### （2）给cobbler添加openstack-newton的repo源

cobbler repo add \

--name=openstack-newton \

--mirror=https://mirrors.aliyun.com/centos/7.3.1611/cloud/x86\_64/openstack-newton/ \

--arch=x86\_64 --breed=yum



repo=构建yum仓库

add=添加

--name=源的名字

--mirror=源的地址

### （3）刷新同步repo

[root@linux-node1 ~]# cobbler reposync

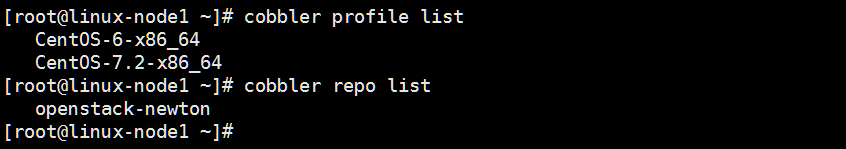
重置repo，使配置文件生效，cobbler会将网站下面所有的包下载到本地，并且帮我们创建repo文件(/etc/yum.repos.d)



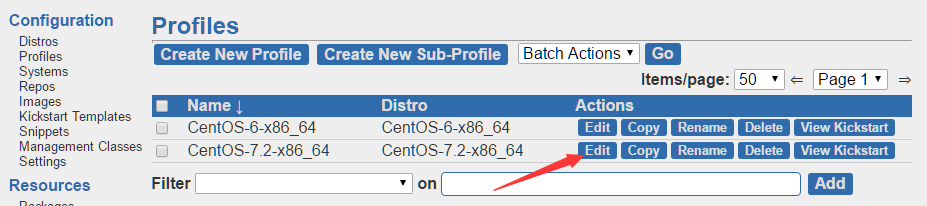
同步完成，安装路径如下

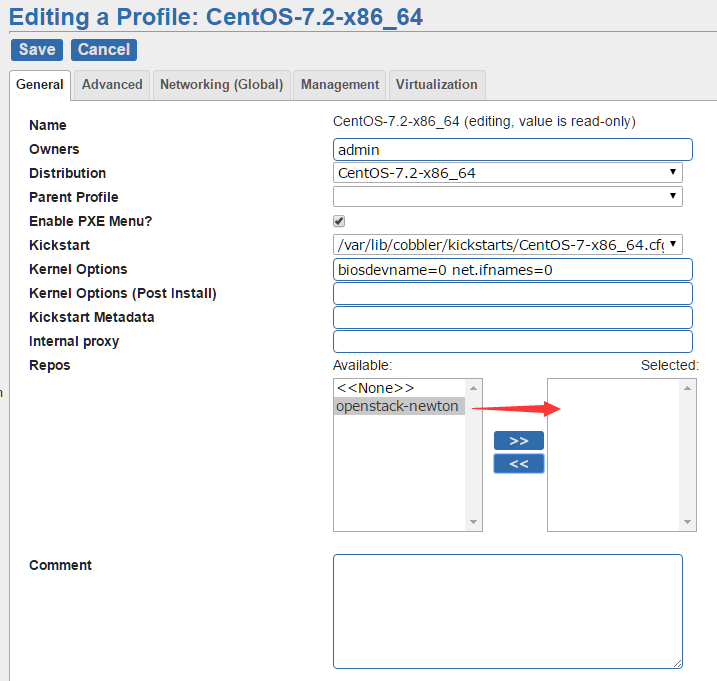


### （4）添加repo到对应的profile



方法一：cobbler web



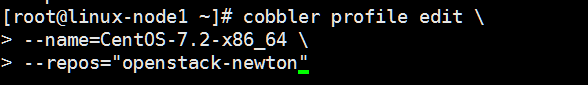


方法二：命令行

cobbler profile edit \

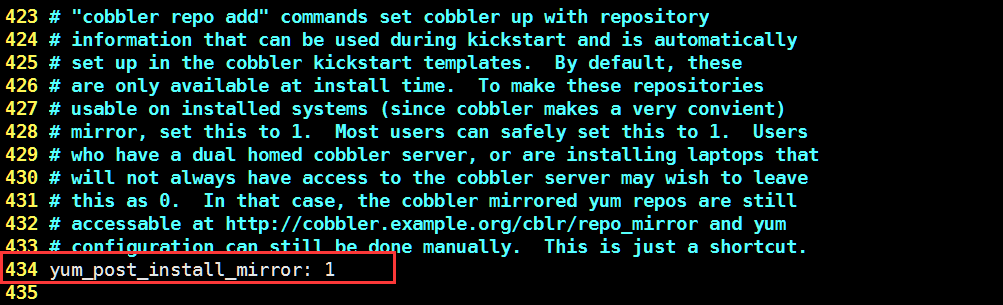
--name=CentOS-7.2-x86\_64 \

--repos="openstack-newton"



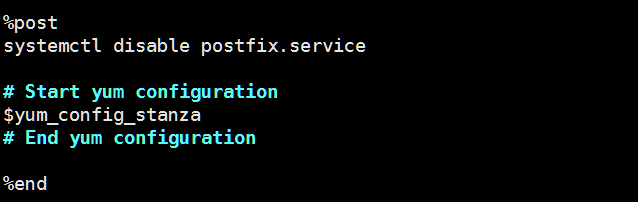
默认情况下，在安装系统的时候会默认安装openstack-newton

配置在cobbler配置文件/etc/cobbler/settings



### （5）修改kickstart文件

我们在kickstarts的配置文件中加入$yum\_config\_stanaz,这样在安装的时候，会自动调解yum仓库（具体调解什么我也不太清楚，网上是这样写的，应该有好处）



赵班长给得cfg配置文件里就是写好了的

### （6）添加定时任务，定期同步repo

如果在生产环境中，我们同步了epel源，那我们有必要每天晚上对repo定期做同步

[root@linux-node1 ~]# echo "00 00 \* \* \* /usr/bin/cobbler reposync --tries=3 --no-file" >>/var/spool/cron/root

## 14、自动化的一套流程

我们可以按照需求，对指定mac地址的服务器进行设置

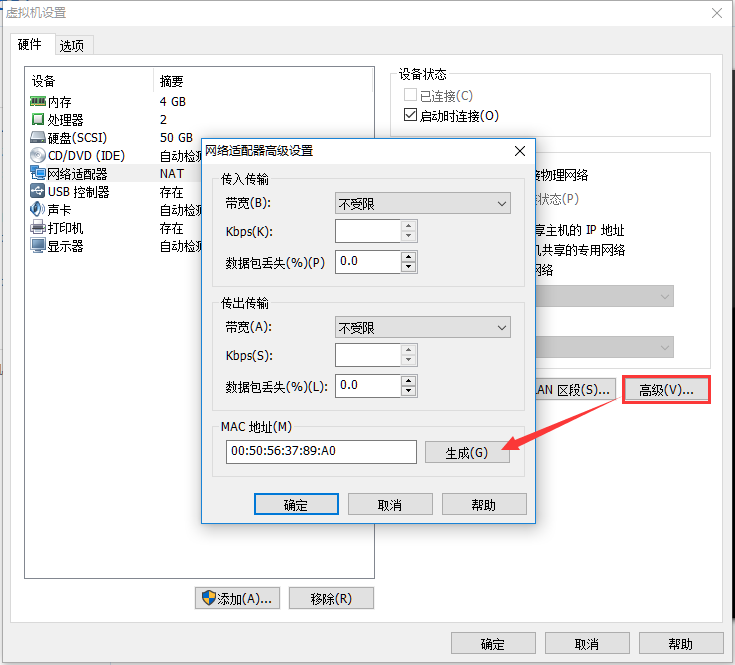
* profile
* 操作系统
* 分区
* 预分配的IP地址
* 主机名
* 子网
* 网关
* dns
* 角色

规划如下:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IP地址 | 主机名 | 掩码 | 网关 | DNS |
| 192.168.56.12 | Linux-node1 | 255.255.255.0 | 192.168.56.2 | 182.168.56.2 |

### （1）查看mac地址

模拟环境mac地址生产办法00:50:56:37:89:A0



### （2）cobbler system定制设置

cobbler system add \

--name=Linux-node1.com \

--mac=00:50:56:37:89:A0 \

--profile=CentOS-7.2-x86\_64 \

--ip-address=192.168.56.12 \

--subnet=255.255.255.0 \

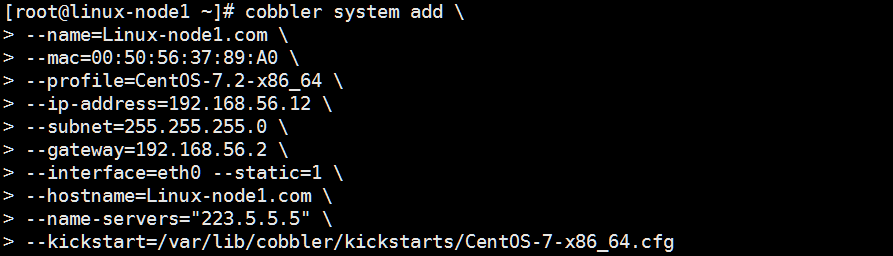
--gateway=192.168.56.2 \

--interface=eth0 --static=1 \

--hostname=Linux-node1.com \

--name-servers="223.5.5.5" \

--kickstart=/var/lib/cobbler/kickstarts/CentOS-7-x86\_64.cfg



### （3）检查，并刷新，使配置生效

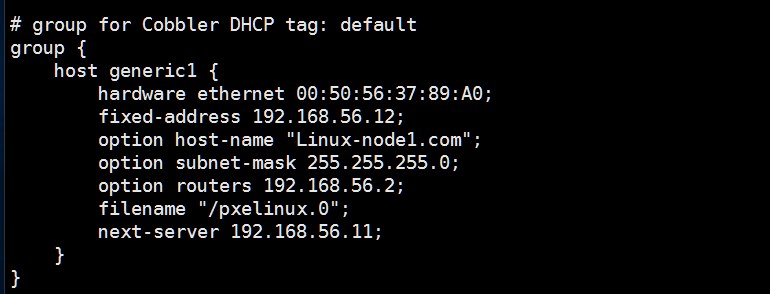
[root@linux-node1 ~]# cobbler system list

Linux-node1.com

[root@linux-node1 ~]# cobbler sync

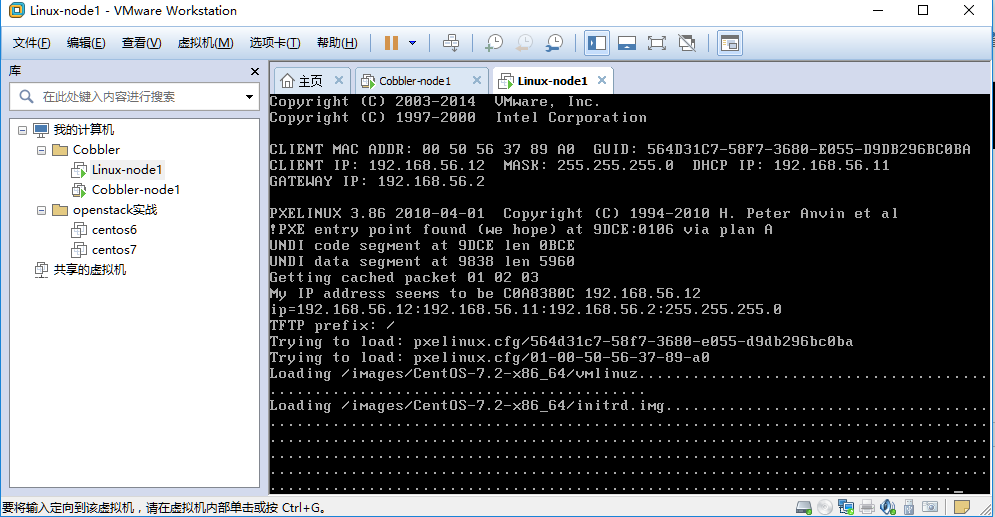
### （4）查看dhcp配置文件，里边已经生成了一个自定义主机配置

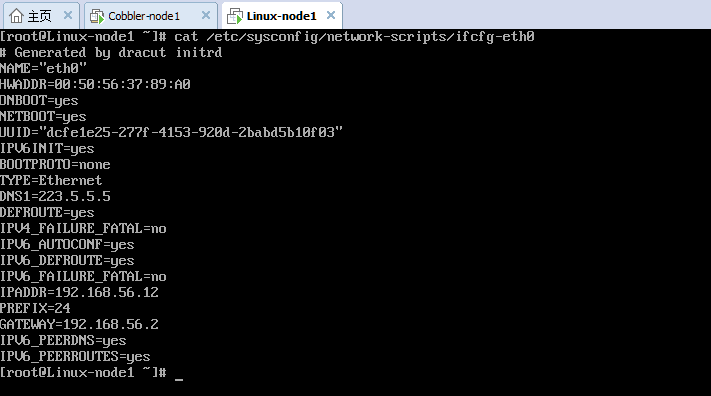
cat /etc/dhcp/dhcpd.conf



### （5）创建虚拟机，cobbler会根据mac地址自动识别安装，不会出现cobbler选择系统界面

注意，mac地址必须与自定义配置中的mac地址一致





安装完成！

cobbler大部分是使用脚本实现的

脚本路径如下：

[root@linux-node1 ~]# cd /var/lib/cobbler/snippets/

