Cobbler Installation Part 2

[**Introduction:**](#_wmgubqaoaa0p) **1**

[**Installing the Cobbler Client Kickstart file:**](#_5thkltbgkada) **5**

[**Customizing the kickstart file for your Cobbler Client:**](#_1vatyv4emez6) **6**

[**Getting the MAC Address of the Cobbler Client:**](#_3mqvci324hjq) **8**

[**Defining a host in the Cobbler Server:**](#_a5j3rwhg33b8) **9**

[**Create a Cobbler profile for your Cobbler Client:**](#_ch1vvokluj7i) **10**

[**Create the system for your Cobbler Client:**](#_priwyiqmj190) **10**

[**Restart Cobbler to sync the system:**](#_nmua4tkf5o7n) **10**

[**Check that you have a system defined in Cobbler:**](#_ltvvv0gt9wr2) **12**

[**APPENDIX A:**](#_sco9gvrluis4) **15**

[Writing a script:](#_h3aol77cu3t5) 15

[Writing a script using variables:](#_9uv3j6a0hts5) 15

[Another script with more variables:](#_8nri8ihry6jt) 16

[**APPENDIX: B**](#_t89wq4ii3igc) **17**

[Full kickstart file for Cobbler Client:](#_nbb523g2xit) 17

# Introduction:

In the first part of installing Cobbler, we installed the basic pieces for the Cobbler server. Please see Class 3 - Installing Cobbler Part 1. As in Part 1, this documentation is based on the article: <https://www.ibm.com/developerworks/library/l-cobbler>.

The last step in Part 1 was to import a Linux distribution into Cobbler. If all went well, you should now be able to run:

|  |
| --- |
| cobbler distro report |

With an output that is similar to:

|  |
| --- |
| Name : CentOS7-x86\_64  Architecture : x86\_64  TFTP Boot Files : {}  Breed : redhat  Comment :  Fetchable Files : {}  Initrd : /var/www/cobbler/ks\_mirror/CentOS7-x86\_64/images/pxeboot/initrd.img  Kernel : /var/www/cobbler/ks\_mirror/CentOS7-x86\_64/images/pxeboot/vmlinuz  Kernel Options : {}  Kernel Options (Post Install) : {}  Kickstart Metadata : {'tree': 'http://@@http\_server@@/cblr/links/CentOS7-x86\_64'}  Management Classes : []  OS Version : rhel7  Owners : ['admin']  Red Hat Management Key : <<inherit>>  Red Hat Management Server : <<inherit>>  Template Files : {} |

Cobbler is a hierarchical system of:

|  |
| --- |
| distributions -> profiles -> systems |

* Distributions refer to the Linux distributions such as CentOS7
* Profiles include a distribution, a kickstart file, and possibly repositories, plus other data such as more specific kernel parameters.
* Systems are the machine to be provisioned. It points to a profile or an image and contains information about IP and MAC addresses, power management (address, credentials, type), and more specialized data.

|  |
| --- |
| The Red Hat **Kickstart** installation methodis used primarily (but not exclusively) by theRed Hat operating system to automatically perform unattended operating system installation and configuration. The **Kickstart** method uses a configuration file commonly referring to as the “**kickstart file**” to manage the unattended installation |

Notice that importing the distribution also created a profile:

|  |
| --- |
| cobbler profile report |

|  |
| --- |
| Name : CentOS7-x86\_64  Architecture : x86\_64  TFTP Boot Files : {}  Breed : redhat  Comment :  Fetchable Files : {}  Initrd : /var/www/cobbler/ks\_mirror/CentOS7-x86\_64/images/pxeboot/initrd.img  Kernel : /var/www/cobbler/ks\_mirror/CentOS7-x86\_64/images/pxeboot/vmlinuz  Kernel Options : {}  Kernel Options (Post Install) : {}  Kickstart Metadata : {'tree': 'http://@@http\_server@@/cblr/links/CentOS7-x86\_64'}  Management Classes : []  OS Version : rhel7  Owners : ['admin']  Red Hat Management Key : <<inherit>>  Red Hat Management Server : <<inherit>>  Template Files : {} |

The commands to work with distributions are:

|  |
| --- |
| cobbler distro “command” [options]  For example:  cobbler distro report  Try running:  cobbler distro help |

You can probably guess how to work with profiles:

|  |
| --- |
| cobbler profile “command” [options]  For example:  cobbler profile report  Try running:  cobbler profile help |

And systems:

|  |
| --- |
| cobbler system “command” [options]  For example:  cobbler system report  Try running:  cobbler system help |

# Installing the Cobbler Client Kickstart file:

Before we can add a system or host to Cobbler we need to create a kickstart file. Whenever you install CentOS (or Red Hat), the installation process creates a kickstart file in the /root directory called **anaconda-ks.cfg.**  I started with that file and modified it (significantly) to create the kickstart file you will use in the class. The file is available as a text file in Google drive folder ([htcxxx.ks](https://drive.google.com/open?id=1nJloNvOEuVwqn1ji0fHxv4ZUDw-ZWxl9TFzj7BcBvhA)) or fromthe main server in the room at ([http://htc180.najah.edu/software/AnNajah-Files/htcXXX.ks](http://172.16.9.180/software/AnNajah-Files/htcXXX.ks)). The file is included at the bottom of this document an an appendix.

This file goes in the directory **/var/lib/cobbler/kickstarts/** with the recommended name htcXXX.ks where htcXXX is the hostname of the Cobbler Client machine.

For example:

|  |
| --- |
| /var/lib/cobbler/kickstarts/htc193.ks |

You can use the following to get the file from class server:

|  |
| --- |
| Be sure to replace **193** with the IP number for your Cobbler Client. |

|  |
| --- |
| cd /var/lib/cobbler/kickstarts  curl -o htcXXX.ks [http://htc180.najah.edu/software/AnNajah-Files/htcXXX.ks](http://172.16.9.180/software/AnNajah-Files/htcXXX.ks)  cp htcXXX.ks **htc193.ks** |

# Customizing the kickstart file for your Cobbler Client:

|  |
| --- |
| WARNING: A kickstart file can look like a very complicated configuration file. We will discuss the file in class. Remember to ask questions. |

Before you can use the file, it has to be configured for your host.

Start by generating an encrypted password exactly how it was done previously:

|  |
| --- |
| openssl passwd -1 |

Open the file (in this example it is htc193.ks) using an editor:

|  |
| --- |
| vi /var/lib/cobbler/htc193.ks |

* Find the line:

|  |
| --- |
| rootpw --iscrypted **You\_Must\_Put\_Encrypted\_Password\_Here** |

And replace **You\_Must\_Put\_Encrypted\_Password\_Here** with the generated encrypted password.

|  |
| --- |
| **NOTE**: The encrypted password is **NOT** enclosed in **quotes**.  **This is very important!** |

Now do the same thing for the line:

|  |
| --- |
| user --name=student --password=**You\_Must\_Put\_Encrypted\_Password\_Here** --iscrypted --gecos="Student" |

* Find the place where the network is defined for Cobbler Client:

|  |
| --- |
| network --bootproto=static --gateway=**GATEWAY** --ip=**IP\_ADDRESS** --nameserver=**DNS\_NAMESERVER** --netmask=**NETMASK** --hostname=**HOSTNAME** --noipv6 |

And replace the following:

|  |
| --- |
| **All field (values) refer to the Cobbler Client**   * + **GATEWAY -** The gateway (172.16.1.2)   + **IP\_ADDRESS** - Cobbler Client Address   + **DNS\_NAMESERVER** - DNS (172.16.0.1)   + **NETMASK** - (255.255.240.0)   + **HOSTNAME** - This is the full host name of the Cobbler Client. For example: htc183.najah.edu |

A few items to notice about the kickstart file:

* **url --url=$tree** and **$SNIPPET('kickstart\_done')** are Cobbler specific directives. You will not find these in the kickstart file created at installation time. Recall that Cobbler uses a hierarchy of distro -> profile -> system. **$tree** is the distro in this hierarchy for your particular system. **$SNIPPET('kickstart\_done')** is used by Cobbler to signify that an installation succeeded in order to prevent the host from reinstalling the operating system on the next reboot.
* **%post%** - Contains commands that are very similar to the ones you used when you installed your first host. You can put almost anything here to configure the host at installation. This means you do not have to repeat the initial host installation steps for this host.
* There are a series of **%keyword%**  **%end%**. These group parts of the kickstart into sections that are used at different times during the installation process.
* This kickstart file disables the firewall. In general, this is a bad practice. We will fix this later during the Puppet configuration.
* You can find all the details about configuring and using kickstart at <https://anaconda-installer.readthedocs.io/en/latest/kickstart.html>.

# Getting the MAC Address of the Cobbler Client:

Before you configure your client, you must know its MAC address and it must be configured to use network boot first.

Turn the power on on the client machine. Interrupt the boot process and enter the BIOS.

|  |
| --- |
| Use F2 to interrupt and enter BIOS setup  Go to Settings -> System Configuration -> Integrated NIC  On the right side of screen select Enable with PXE  Click Apply  Go to Settings -> General -> Boot Sequence  Disable Onboard or USB Floppy Drive  Disable Onboard ot USB CD-ROM Drive  Set boot order to:  Inboard Network Controller  USB Drive  ST31000524A ( The internal hard drive)  Click Apply  Exit |

Now the client will try to boot using the network. During this process it will display the MAC address, Write this down on piece of tape and stick the tape to the side of the machine.

|  |
| --- |
| NOTE: The MAC address should be in ‘:’ format:  For example: 00:AA:BB:CC:DD:EE |

# Defining a host in the Cobbler Server:

Now that there is a kickstart file in **/var/lib/cobbler/kickstarts**, the client has been set to **network boot** and you know the client’s **MAC address**, we can configure **Cobbler** to install the operating when **Client** reboots. This called an **unattended installation**.

|  |
| --- |
| For the commands below, it is important to **replace htcXXX** with the name of  **your Cobbler Client.** |

|  |
| --- |
| In the commands that follow, you must insert the values for your Cobbler Client:   * **htcXXX** - Cobbler Client name * **MAC** - Cobbler Client MAC Address * **IP\_Address** - IP address of the Cobbler Client * **GATEWAY** - 172.16.1.2 * **NETMASK** - 255.255.240.0 |

# Create a Cobbler profile for your Cobbler Client:

|  |
| --- |
| cobbler profile add --name=htcXXX --kickstart=/var/lib/cobbler/kickstarts/htcXXX.ks --parent=CentOS7-x86\_64 |

# Create the system for your Cobbler Client:

|  |
| --- |
| cobbler system add --name=htcXXX --profile=htcXXX --interface=eth0 --mac=MAC --ip-address=IP\_ADDRESS --gateway=GATEWAY --netmask=NETMASK --static=1 --dns-name=htcXXX.najah.edu |

# Restart Cobbler to sync the system:

|  |
| --- |
| systemctl restart cobblerd  sleep 10  cobbler sync  systemctl restart xinetd |

You should output similar to (this output is long and spans 2 pages):

|  |
| --- |
| task started: 2018-03-12\_114929\_sync  task started (id=Sync, time=Mon Mar 12 11:49:29 2018)  running pre-sync triggers  cleaning trees  removing: /var/www/cobbler/images/CentOS7-x86\_64  removing: /var/lib/tftpboot/pxelinux.cfg/default  removing: /var/lib/tftpboot/pxelinux.cfg/01-18-03-73-18-b2-ad  removing: /var/lib/tftpboot/grub/images  removing: /var/lib/tftpboot/grub/efidefault  removing: /var/lib/tftpboot/grub/01-18-03-73-18-B2-AD  removing: /var/lib/tftpboot/images/CentOS7-x86\_64  removing: /var/lib/tftpboot/s390x/profile\_list  copying bootloaders  copying: /usr/share/syslinux/pxelinux.0 -> /var/lib/tftpboot/pxelinux.0  copying: /usr/share/syslinux/menu.c32 -> /var/lib/tftpboot/menu.c32  copying: /usr/share/syslinux/memdisk -> /var/lib/tftpboot/memdisk  copying distros to tftpboot  copying files for distro: CentOS7-x86\_64  trying hardlink /var/www/cobbler/ks\_mirror/CentOS7-x86\_64/images/pxeboot/vmlinuz -> /var/lib/tftpboot/images/CentOS7-x86\_64/vmlinuz  trying hardlink /var/www/cobbler/ks\_mirror/CentOS7-x86\_64/images/pxeboot/initrd.img -> /var/lib/tftpboot/images/CentOS7-x86\_64/initrd.img  copying images  generating PXE configuration files  generating: /var/lib/tftpboot/pxelinux.cfg/01-18-03-73-18-b2-ad  generating: /var/lib/tftpboot/grub/01-18-03-73-18-B2-AD  generating PXE menu structure  copying files for distro: CentOS7-x86\_64  trying hardlink /var/www/cobbler/ks\_mirror/CentOS7-x86\_64/images/pxeboot/vmlinuz -> /var/www/cobbler/images/CentOS7-x86\_64/vmlinuz  trying hardlink /var/www/cobbler/ks\_mirror/CentOS7-x86\_64/images/pxeboot/initrd.img -> /var/www/cobbler/images/CentOS7-x86\_64/initrd.img  Writing template files for CentOS7-x86\_64  rendering DHCP files  rendering DNS files  rendering TFTPD files  generating /etc/xinetd.d/tftp  processing boot\_files for distro: CentOS7-x86\_64  cleaning link caches  running post-sync triggers  running python triggers from /var/lib/cobbler/triggers/sync/post/\*  running python trigger cobbler.modules.sync\_post\_restart\_services  running: service dnsmasq restart  received on stdout:  received on stderr: Redirecting to /bin/systemctl restart dnsmasq.service  running shell triggers from /var/lib/cobbler/triggers/sync/post/\*  running python triggers from /var/lib/cobbler/triggers/change/\*  running python trigger cobbler.modules.scm\_track  running shell triggers from /var/lib/cobbler/triggers/change/\*  \*\*\* TASK COMPLETE \*\*\* |

# Check that you have a system defined in Cobbler:

|  |
| --- |
| cobbler system report --name=htcXXX |

The output should be similar to (this output is long and spans 2 pages):

|  |
| --- |
| Name : htc182  TFTP Boot Files : {}  Comment :  Enable gPXE? : <<inherit>>  Fetchable Files : {}  Gateway : 172.16.1.2  Hostname :  Image :  IPv6 Autoconfiguration : False  IPv6 Default Device :  Kernel Options : {}  Kernel Options (Post Install) : {}  Kickstart : <<inherit>>  Kickstart Metadata : {}  LDAP Enabled : False  LDAP Management Type : authconfig  Management Classes : <<inherit>>  Management Parameters : <<inherit>>  Monit Enabled : False  Name Servers : []  Name Servers Search Path : []  Netboot Enabled : True  Owners : <<inherit>>  Power Management Address :  Power Management ID :  Power Management Password :  Power Management Type : ipmitool  Power Management Username :  Profile : htc182  Internal proxy : <<inherit>>  Red Hat Management Key : <<inherit>>  Red Hat Management Server : <<inherit>>  Repos Enabled : False  Server Override : <<inherit>>  Status : production  Template Files : {}  Virt Auto Boot : <<inherit>>  Virt CPUs : <<inherit>>  Virt Disk Driver Type : <<inherit>>  Virt File Size(GB) : <<inherit>>  Virt Path : <<inherit>>  Virt PXE Boot : 0  Virt RAM (MB) : <<inherit>>  Virt Type : <<inherit>>  Interface ===== : eth0  Bonding Opts :  Bridge Opts :  CNAMES : []  InfiniBand Connected Mode : False  DHCP Tag :  DNS Name : htc182.najah.edu  Per-Interface Gateway :  Master Interface :  Interface Type :  IP Address : 172.16.9.182  IPv6 Address :  IPv6 Default Gateway :  IPv6 MTU :  IPv6 Prefix :  IPv6 Secondaries : []  IPv6 Static Routes : []  MAC Address : 00:aa:00:bb:00:cc  Management Interface : False  MTU :  Subnet Mask : 255.255.240.0  Static : True  Static Routes : []  Virt Bridge : |

Or to see just the relevant information:

|  |
| --- |
| export host=htcXXX  cobbler system report --name=$host | egrep 'Name|Gateway|Hostname|Profile|IP Address|MAC Address|Subnet Mask' |

with the more simplified output:

|  |
| --- |
| Name : htc192  Gateway : 172.16.1.2  Hostname :  Name Servers : []  Name Servers Search Path : []  Profile : htc192  DNS Name : htc192.najah.edu  Per-Interface Gateway :  IP Address : 172.16.9.192  IPv6 Default Gateway :  MAC Address : 00:aa:00:bb:00:cc  Subnet Mask : 255.255.240.0 |

Now if everything went well, you can turn the power on on your client node and it will install the operating system, do some minimal configuration and reboot. Make sure to turn on the monitor during the installation so you can watch the magic.

# APPENDIX A:

## Writing a script:

We seem to use the command:

|  |
| --- |
| systemctl restart cobblerd  sleep 10  cobbler sync  systemctl restart xinetd |

Can you create a script that issues these commands? How do you make a script executable? Where can you store the script so that root can find it without using the full path?

## Writing a script using variables:

Previously we used:

|  |
| --- |
| export host=htcXXX  cobbler system report --name=$host | egrep 'Name|Gateway|Hostname|Profile|IP Address|MAC Address|Subnet Mask' |

In these commands:

|  |
| --- |
| export host=htcXX |

defines the variable host to have the value htcXXX.

Try the following:

|  |
| --- |
| export host=htcXXX  echo $host  export host=htc192  echo $host |

Put the commands above in script and execute the script.

## Another script with more variables:

Use **“export”** and create a script to issue the commands below. You want to use the **export** command to define the variables:

* **htcXXX**
* **GATEWAY**
* **NETMASK**

|  |
| --- |
| #  # The cobble commands are one line  #  echo `date`" Creating new profile: htcXXX"  cobbler profile add --name=htcXXX --kickstart=/var/lib/cobbler/kickstarts/htcXXX.ks --parent=CentOS7-x86\_64  echo `date`" Creating new system: htcXXX"  cobbler system add --name=htcXXX --profile=htcXXX --interface=eth0 --mac=MAC --ip-address=IP\_ADDRESS --gateway=GATEWAY --netmask=NETMASK --static=1 --dns-name=htcXXX.najah.edu  cobbler sync |

|  |
| --- |
| Ask for help if this is confusing.  Ask for help if anything is confusing. |

# APPENDIX: B

## Full kickstart file for Cobbler Client:

## 

|  |
| --- |
| # Kickstart file create: Tue Mar 6 07:32:48 MST 2018 for htcXXX.najah.edu  # platform=x86, AMD64, or Intel EM64T  # version=DEVEL  #------------------------------------------------------------------------  # Install OS instead of upgrade  #  install  #------------------------------------------------------------------------  # Keyboard layout  #  keyboard 'us'  #------------------------------------------------------------------------  # Root password:  # openssl passwd -1  # IMPORTANT: Encrypted password DOES NOT have quotes  rootpw --iscrypted You\_Must\_Put\_Encrypted\_Password\_Here  #  #------------------------------------------------------------------------  # Create the student account  # IMPORTANT: Encrypted password DOES NOT have quotes #  user --name=student --password=You\_Must\_Put\_Encrypted\_Password\_Here --iscrypted --gecos="Student"  #  #------------------------------------------------------------------------  # System language  #  lang en\_US  #------------------------------------------------------------------------  # Firewall configuration  #  firewall --disabled  #  #------------------------------------------------------------------------  # System authorization information  #  auth --enableshadow --passalgo=sha512  #  #------------------------------------------------------------------------  # Use text mode install  #  text  firstboot --disable  #  #------------------------------------------------------------------------  # SELinux configuration  #  selinux --disabled  #  #------------------------------------------------------------------------  # Installation media:  #  # This is a Cobbler specific directive  # See: Creating Profiles at https://www.ibm.com/developerworks/library/l-cobbler/  # Lots of details: http://cheetahtemplate.org/  # The souce for Cobbler is a special directive that Cobbler will fill in from the definition for this host  #  url --url=$tree  #  #------------------------------------------------------------------------  # Network information  #  # Replace the values in the line below with the ones for your client  #  network --bootproto=static --gateway=GATEWAY --ip=IP\_ADDRESS --nameserver=DNS\_NAMESERVER --netmask=NETMASK --hostname=HOSTNAME --noipv6  #  #------------------------------------------------------------------------  # Reboot after installation  #  reboot  #  #------------------------------------------------------------------------  # System timezone  #  timezone Asia/Jerusalem  #------------------------------------------------------------------------  # System bootloader configuration  #  bootloader --append=" crashkernel=auto" --location=mbr --boot-drive=sda  #------------------------------------------------------------------------  # Partition clearing information  autopart --type=lvm  clearpart --all --initlabel --drives=sda  #------------------------------------------------------------------------  # System services  #  services --enabled="chronyd"  #------------------------------------------------------------------------  # This defines the installation packages  # This is a minimum installation  #  %packages  @core  chrony  kexec-tools  %end  #------------------------------------------------------------------------  # Define password policies for the system  # See: https://anaconda-installer.readthedocs.io/en/latest/  # https://anaconda-installer.readthedocs.io/en/latest/kickstart.html  #  %anaconda  pwpolicy root --minlen=6 --minquality=50 --notstrict --nochanges --notempty  pwpolicy user --minlen=6 --minquality=50 --notstrict --nochanges --notempty  pwpolicy luks --minlen=6 --minquality=50 --notstrict --nochanges --notempty  %end  #  #-----------------------------------------------------------------------------------  # Post installation commands  # After the installation completes, run these commands  # This looks a lot like how we set up the first machine  #  %post --log=/root/ks-post.log  curl -o /etc/hosts http://172.16.9.180/software/AnNajah-Files/hosts  curl -o /root/bootstrap.tar.gz http://htc180.najah.edu/software/bootstrap/bootstrap.tar.gz  tar -C /root -xzvf /root/bootstrap.tar.gz  /root/install/InstallRepos  /root/install/InstallPackages  yum clean all  yum -y update  #  # Make sure that the ssh port is open on the firewall  #  firewall-cmd --add-port=22/tcp --permanent  #  # Turn off selinux  # You did this by editing the file  # This is how you can do it from the command line  #  /usr/bin/sed s/SELINUX=enforcing/SELINUX=disabled/ /etc/selinux/config > /tmp/config; mv /tmp/config /etc/selinux/config  #  # Install lsb - Linux Standard Base  # See: https://en.wikipedia.org/wiki/Linux\_Standard\_Base  #  yum -y install \*lsb\*  #  # This is a Cobbler specific directive  # See: Creating Profiles at https://www.ibm.com/developerworks/library/l-cobbler/  # Lots of details: http://cheetahtemplate.org/  #  $SNIPPET('kickstart\_done')  reboot  %end |