Class 10 - A Solution to Class 9 Problem

[**Introduction:**](#_pvli0wro0pgu) **1**

[What the hints tell us:](#_wz4m17xbq0vt) 2

[Files Installed during installation:](#_5n1pglmgcpuo) 2

[**Puppet Servers run Web Server:**](#_u7hnriv8zp1x) **3**

[Procedure:](#_y28p4ysnzj8b) 3

[Puppet Module to Manage Host Keys:](#_hf3boqjsu1kd) 4

[**Storing and Restoring Puppet Client certificates:**](#_qlsrf52l7hx9) **7**

[**CentOS installation to previous state:**](#_hvzw8cx4gfhc) **8**

[**Appendix A:**](#_r97yko6at9c0) **9**

# Introduction:

|  |
| --- |
| Throughput this document I am using:  **Puppet Server** is **htc191**  **Puppet Client** is **htc192** |

This is my solution to the problem posed in Class 9. It is not necessarily the only solution and it is not a complete solution but it should work using the tools we have learned so far. We will address the problem one step at a time and then try to put it all together.

|  |
| --- |
| Project:   * Create a Puppet module to install the host ssh keys. * Create a means to save the Puppet Client certificates. * Create a means to reinstall the operating system such that:   + Puppet Client is automatically configured at installation time.   + The Puppet Client certificates do not change after a reinstallation. * **Goal:** The Puppet Client should be able to go through a reinstallation of the operating system and be identically configured to how it was prior to the reinstallation. This should all be done without ever logging into the Puppet Client. * **Hint:** There are some files that are already installed at installation time.   How is that done? * **Further hint:** Puppet Servers run a web server.   You may use any source you like to try to address this problem. |

## What the hints tell us:

We will take a look at the hints and see how they might help.

## Files Installed during installation:

During the **USB** installation and the **kickstart** installation there are a number of additional files that were installed. In a **production** system, most of these files should be **moved** to **Puppet** and the **kickstart** file made as simple (**minimal**) as possible.

Open your kickstart file (**/var/lib/cobber/kickstarts/**) and look for the lines:

|  |
| --- |
| 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  curl -o /root/cobbler-files.tar.gz http://htc180.najah.edu/software/bootstrap/cobbler-files.tar.gz  mkdir -p /root/install/startup  tar -C /root/install -xzvf /root/bootstrap.tar.gz  tar -C /root/install -xzvf /root/cobbler-files.tar.gz  /root/install/startup/InstallRepos  /root/install/startup/InstallPackages |

This part of the kickstart file:

* Installs **/etc/hosts**
* Gets two tar files from the server: **bootstrap.tar.gz** and **cobbler-files.tar.gz**
* **bootstrap.tar.gz** contains two scripts:
  + **InstallRepos** - Installs the **repo** files (configuration files) so that packages are installed from the **local server**.
  + **InstallPackages** - Installs a number of initial packages.

|  |
| --- |
| * We only used **InstallRepos** because we wanted to use the **repos** on the **class server.** * We never used **cobbler-files.tar.gz** it could be **removed** from the kickstart file. * The script **InstallPackages** is a number of standard package installations that are easily moved to Puppet. |

All of these procedures can be moved out of the kickstart file and into **Puppet**. At installation time the **Puppet Client** only needs:

|  |
| --- |
| * **Puppet installed and enabled** * **/etc/puppetlabs/puppet/puppet.conf** configured * **systemctl enable puppet-agent** |

**For now** I will leave these line in the **kickstart file**, but will try to explain how to move them out of the **kickstart** file and into **Puppet** in **Appendix A**.

# Puppet Servers run Web Server:

When you **installed Puppet Server**, a **web serve**r was installed. We can use that web server to store and then transfer the **Puppet Client ssl** files during a system installation. By default the web server will read files in **/var/www/html**. If you make a **subdirectory** under **/var/www/html** you can use it for files that can be **transferred** using **curl** during the **installation process**.

Recall that the **Puppet Client ssl** files are store in:

|  |
| --- |
| /etc/puppetlabs/puppet/ssl/ |

If we **save** these files and then **restore** them during a system installation, the **Puppet Client** will be returned to its original state. In other words, it will have **puppet configured** and have its previous **certificate**. We will use the **tar** command to create a copy of this directory.

|  |
| --- |
| In **Linux**, the name of the **tar command** is short for **tape archiving**. A common use for **tar** is to **combine a few files** into a **single file**, for easy storage and distribution. See: [**https://kb.iu.edu/d/acfi**](https://kb.iu.edu/d/acfi) |

On the **Puppet Client** create a **tar file** of the **ssl directory**:

|  |
| --- |
| root> cd /etc/puppetlabs/puppet  root> tar -czvf /tmp/htc192-ssl.tar.gz ssl/ |

We want to put this file on the **Puppet Server** in a directory accessible by the **web server.** On the **Puppet Server,** make a directory where you can **store** the **Puppet Client ssl files** and then **copy** the **ssl tar file** from the **Puppet Client** to the **Puppet Server**.

|  |
| --- |
| root> mkdir /var/www/html/htc192  root> cd /var/www/html/htc192  root> sftp htc192:/tmp/htc192-ssl.tar.gz |

Now the **Puppet Server** can get the **Puppet Client** **ssl files** (keys) using:

|  |
| --- |
| root> curl -o /tmp/htc192-ssl.tar.gz http://htc191.najah.edu/htc192/htc192-ssl.tar.gz |

|  |
| --- |
| We will look at the needed changes to the kickstart file to install these in the section **Storing and Restoring Puppet Client certificates.** |

## Puppet Module to Manage Host Keys:

|  |
| --- |
| **Puppet modules** are **created** on the **Puppet Server**. |

**Start** by creating the **module skeleton**:

|  |
| --- |
| root> cd /etc/puppetlabs/code/environments/production/modules  root> puppet module generate htcclass-host\_ssh\_keys  root> mkdir host\_ssh\_keys/files |

Now get the **Puppet Client** **host ssh keys** using **sftp** and store them in **host\_ssh\_keys/files**:

|  |
| --- |
| root> cd host\_ssh\_keys/files  root> sftp htc192 |

Sample session:

|  |
| --- |
| **[root@htc191 files]# sftp htc192**  root@htc192's password:  Connected to htc192.  **sftp> cd /etc/ssh**  **sftp> get ssh\_host\_\***  Fetching /etc/ssh/ssh\_host\_ecdsa\_key to ssh\_host\_ecdsa\_key  /etc/ssh/ssh\_host\_ecdsa\_key 100% 227 138.2KB/s 00:00  Fetching /etc/ssh/ssh\_host\_ecdsa\_key.pub to ssh\_host\_ecdsa\_key.pub  /etc/ssh/ssh\_host\_ecdsa\_key.pub 100% 162 87.7KB/s 00:00  Fetching /etc/ssh/ssh\_host\_ed25519\_key to ssh\_host\_ed25519\_key  /etc/ssh/ssh\_host\_ed25519\_key 100% 387 270.4KB/s 00:00  Fetching /etc/ssh/ssh\_host\_ed25519\_key.pub to ssh\_host\_ed25519\_key.pub  /etc/ssh/ssh\_host\_ed25519\_key.pub 100% 82 49.6KB/s 00:00  Fetching /etc/ssh/ssh\_host\_rsa\_key to ssh\_host\_rsa\_key  /etc/ssh/ssh\_host\_rsa\_key 100% 1679 982.7KB/s 00:00  Fetching /etc/ssh/ssh\_host\_rsa\_key.pub to ssh\_host\_rsa\_key.pub  /etc/ssh/ssh\_host\_rsa\_key.pub 100% 382 237.2KB/s 00:00 |

|  |
| --- |
| When you **transfer the files**, they will **not be readable** by the **Puppet Server**. The **Puppet Server** runs as the user **puppet** but the files are **owned** by **root** and **protected.** |

:

|  |
| --- |
| root> ls -al |

|  |
| --- |
| [root@htc191 files]# ls -l  -rw-r----- 1 root root 227 Apr 1 15:52 ssh\_host\_ecdsa\_key  -rw-r--r-- 1 root root 162 Apr 1 15:52 ssh\_host\_ecdsa\_key.pub  -rw-r----- 1 root root 387 Apr 1 15:52 ssh\_host\_ed25519\_key  -rw-r--r-- 1 root root 82 Apr 1 15:52 ssh\_host\_ed25519\_key.pub  -rw-r----- 1 root root 1679 Apr 1 15:52 ssh\_host\_rsa\_key  -rw-r--r-- 1 root root 382 Apr 1 15:52 ssh\_host\_rsa\_key.pub |

We can fix this by changing the ownership to puppet:

|  |
| --- |
| root> chown puppet.puppet \* |

Now:

|  |
| --- |
| [root@htc191 files]# ls -l  total 24  -rw-r----- 1 puppet puppet 227 Apr 1 15:52 ssh\_host\_ecdsa\_key  -rw-r--r-- 1 puppet puppet 162 Apr 1 15:52 ssh\_host\_ecdsa\_key.pub  -rw-r----- 1 puppet puppet 387 Apr 1 15:52 ssh\_host\_ed25519\_key  -rw-r--r-- 1 puppet puppet 82 Apr 1 15:52 ssh\_host\_ed25519\_key.pub  -rw-r----- 1 puppet puppet 1679 Apr 1 15:52 ssh\_host\_rsa\_key  -rw-r--r-- 1 puppet puppet 382 Apr 1 15:52 ssh\_host\_rsa\_key.pub |

Now we want to modify the init.pp file to install the ssh keys. The class we need is:

|  |
| --- |
| class host\_ssh\_keys {  file {"/etc/ssh/ssh\_host\_ecdsa\_key":  ensure => file,  replace => true,  owner => 'root',  group => 'root',  mode => "0600",  source => "puppet:///modules/host\_ssh\_keys/ssh\_host\_ecdsa\_key",  }  file {"/etc/ssh/ssh\_host\_ecdsa\_key.pub":  ensure => file,  replace => true,  owner => 'root',  group => 'root',  mode => "0644",  source => "puppet:///modules/host\_ssh\_keys/ssh\_host\_ecdsa\_key.pub",  }  file {"/etc/ssh/ssh\_host\_ed25519\_key":  ensure => file,  replace => true,  owner => 'root',  group => 'root',  mode => "0600",  source => "puppet:///modules/host\_ssh\_keys/ssh\_host\_ed25519\_key",  }  file {"/etc/ssh/ssh\_host\_ed25519\_key.pub":  ensure => file,  replace => true,  owner => 'root',  group => 'root',  mode => "0644",  source => "puppet:///modules/host\_ssh\_keys/ssh\_host\_ed25519\_key.pub",  }  file {"/etc/ssh/ssh\_host\_rsa\_key":  ensure => file,  replace => true,  owner => 'root',  group => 'root',  mode => "0600",  source => "puppet:///modules/host\_ssh\_keys/ssh\_host\_rsa\_key",  }  file {"/etc/ssh/ssh\_host\_rsa\_key.pub":  ensure => file,  replace => true,  owner => 'root',  group => 'root',  mode => "0644",  source => "puppet:///modules/host\_ssh\_keys/ssh\_host\_rsa\_key.pub",  }  } |

Now add this class to the list of classes in the module  **role/manifest/init.pp** for the **Puppet Client:**

|  |
| --- |
| #  # htcxxx role  #  class role::htcxxx inherits role {  notice("Top level role for htcxxx")  include motd  include autofs  include host\_ssh\_keys  } |

|  |
| --- |
| A “**feature**” of this solution is that if we expand the **Puppet Server** to manage more hosts, they will get the **same host ssh keys**. This may or may not be the desired effect. We have not discussed how to **separate** files by **host name**. |

# Storing and Restoring Puppet Client certificates:

Most of this section is covered above in **Puppet Servers run Web Server**.

Briefly, we need to save and restore the **Puppet Client**’s files stored in:

|  |
| --- |
| /etc/puppetlabs/puppet/ssl |

We can create a **tar archive** of that directory and **store** it on the **Puppet Server**. The **Puppet Server** is running a **web server**, so we can use that to **transfer** the files during the **installation**. The **kickstart** file that we use for this **installation** should install **puppet agent** and create the **puppet.conf.**

|  |
| --- |
| You should now be using the kickstart file for puppet agent. |

Open the **kickstart** file on the **Cobbler Server**:

|  |
| --- |
| root> cd /var/lib/cobbler/kickstarts/  root> vi htc192-puppet-agent.ks |

Find the lines starting with:

|  |
| --- |
| #  # Install the puppet agent (client)  # |

Then just above the lines:

|  |
| --- |
| systemctl enable puppet  systemctl start puppet |

add the lines:

|  |
| --- |
| curl -o /tmp/htc192-ssl.tar.gz <http://htc191.najah.edu/htc192/htc192-ssl.tar.gz>  tar -C /etc/puppetlabs/puppet /tmp/htc192-ssl.tar.gz |

That section should now look like:

|  |
| --- |
| #  # Install the puppet agent (client)  #  yum -y install puppet-agent  echo "[main]" > /etc/puppetlabs/puppet/puppet.conf  echo " server = htc191.najah.edu" >> /etc/puppetlabs/puppet/puppet.conf  echo "[agent]" >> /etc/puppetlabs/puppet/puppet.conf  **curl -o /tmp/htc192-ssl.tar.gz http://htc191.najah.edu/htc192/htc192-ssl.tar.gz**  **tar -C /etc/puppetlabs/puppet /tmp/htc192-ssl.tar.gz**  systemctl enable puppet  systemctl start puppet |

# CentOS installation to previous state:

We should now have most of the pieces in place to re-install the operating system back to where we started.

Make sure that the kickstart that is being used is the one for puppet-agent. If you are managing your kickstarts with symbolic links, you can:

|  |
| --- |
| root> cd /var/lib/cobbler/kickstarts  root> ln -sf htc192-puppet-agent.ks htc192.ks |

# Appendix A:

Further cleaning of the kickstart file:

Open your **kickstart file** and look for the lines:

|  |
| --- |
| 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  curl -o /root/cobbler-files.tar.gz http://htc180.najah.edu/software/bootstrap/cobbler-files.tar.gz  mkdir -p /root/install/startup  tar -C /root/install -xzvf /root/bootstrap.tar.gz  tar -C /root/install -xzvf /root/cobbler-files.tar.gz  /root/install/startup/InstallRepos  /root/install/startup/InstallPackages |

The first 1 line is used to install the **/etchosts** file on the client. We did this so that we would have the **IP addresses** and **hostnames** for our network. We can fix this by writing a **Puppet module** or **Puppet class** that **manages** the **/etc/hosts** file. For this class, I recommend we leave this.

The next 6 lines download two files from the server and unpack them in the root directory. To fully “Puppetize” the system, we want to look inside the two scripts:

|  |
| --- |
| /root/install/startup/InstallRepos  /root/install/startup/InstallPackages |

We have to check to if there is anything in them that we need during the installation or whether we can put their functionality into Puppet.

The **first script:**

|  |
| --- |
| /root/install/startup/InstallRepos |

installs the **repo files** (repository configuration files in /etc/yum.repo.d).

|  |
| --- |
| For the hosts in our class, it is best to leave this as is. For a production system, you would be able to install these repository configuration form the Internet and then install packages from the Internet. |

InstallPackages can be completely moved to Puppet. If we look at the script, all it does is:

|  |
| --- |
| yum -y install perl emacs bind-utils net-tools nano vim  yum -y install lsof sysstat koan lm\_sensors |

We could create a Puppet module named common\_packages:

|  |
| --- |
| root> cd /etc/puppetlabs/code/environments/production/modules  root> puppet module generate htcclass-common\_packages |

Then the class could be series of Puppet directives such as:

|  |
| --- |
| class common\_packaes{  package {"perl": ensure => installed, }  package {"emacs": ensure => installed, }  …  package {"lm\_sensors": ensure => installed, }  } |

Then this class would be added to the module roles/manifests/init.pp:

|  |
| --- |
| #  # htcxxx role  #  class role::htcxxx inherits role {  notice("Top level role for htcxxx")  include motd  include autofs  include host\_ssh\_keys  include common\_packages  } |

If we do this, we can change this section of the kickstart file to:

|  |
| --- |
| 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  mkdir -p /root/install/startup  tar -C /root/install -xzvf /root/bootstrap.tar.gz  /root/install/startup/InstallRepos |

We have not removed everything from the kickstart file but it is not getting more simple.