

COMP150 Lab 11: DHCP

In this lab, you will install a DHCP server on your Centos VM. We will test it with DHCP clients in a second Centos VM and/or a Windows 7 client.

STEP 1. Install a new Centos client. We can set that up by cloning your existing Centos VM. To clone your VM on a PC:

1. Turn on VMware, but make sure your Centos VM is shut down.
2. Open/select your Centos VM.
3. Go the VM menu and select Manage/Clone.
4. Go through the Wizard using the following settings:
 - a. The current state in the virtual machine
 - b. Create a full clone.
 - c. Name the VM: centos-client
5. Once it's done cloning, you will have a full copy of the original server VM with the same user accounts and configuration that was on your server.

To clone on a Mac you have to copy the file and rename it centos-client.

Once you have the clone

6. Log in to the clone.
7. We need to change one thing to avoid confusion: the name of the system. To do that:
 - a. Open up a terminal.
 - b. Edit `/etc/sysconfig/network` with your preferred editor and `sudo`.
 - c. Change the `HOSTNAME` line to have a name of `centos-client` instead of whatever you have named your server.
 - d. Save the file and exit.
 - e. Next, edit `/etc/hosts` (again using `sudo` and whatever editor you like).
 - f. Change all instances of "hostname" to "centos-client"
 - g. Save the file and exit.
8. These changes won't take effect until the next time you reboot the VM, so reboot it now.
9. When it comes back up, log in and make sure the host name looks ok.
10. **TAKE A SCREENSHOT** of your desktop with an open terminal showing the centos-client name.
11. You can shut the VM down or suspend it for now.

STEP 2. Install a Windows 7 client VM. Get the iso image of win7 sp1 from me or one of your classmates. This will also work with Windows 8 if you prefer. Throughout the lab I will refer to Windows 7 as the generic Windows.

Add the new VM in VMware:

1. Go to the VMware File menu, then select New and Virtual Machine.
2. Do a Typical install.
3. Select an ISO image, and find your Windows 7 Professional ISO.
4. Type in the product key.
5. Be sure to select Windows 7 Professional for the version.
6. Use your normal user name and select a password.
7. Name the VM: windows7-client
8. Leave the disk size at the default.
9. When the Wizard finishes, power on the VM to let Windows install (easy install in VMware).
10. Once it's done, log in and let VMware install the VMware tools for you, after which it will automatically restart the VM.
11. Log in to make sure everything looks ok.
12. **TAKE A SCREENSHOT** of your desktop once logged in.
13. You can shut the VM down or pause it for now.

Now you have all three VMs ready to go.

STEP 3. We need to change the VMware network settings. Specifically, VMware runs its own DHCP server which we want to shut off. To do that:

1. In VMware, go to the Edit menu and select Virtual Network Editor.
2. Highlight VMnet8 (NAT).
3. Uncheck the checkbox: Use a local DHCP service...
4. You can also uncheck the host virtual adapter checkbox since we won't be using that, but you don't have to.
5. Make sure that the Subnet IP is: 192.168.10.0
6. And the Subnet mask is: 255.255.255.0
7. Click Apply, then close the Virtual Network Editor after it's done.

Next we will set up your Centos VM to run DHCP for our virtual network from now on. Power it on, if you haven't already.

STEP 4. Configure the Centos VM with a static IP address. This is necessary since nothing else is going to give it an address automatically.

We need to **stop the NetworkManager service and start the plain network service**.

NetworkManager is great for wireless and mobile clients, but for our server we need something more suited to a static environment. Recall that you can use the `chkconfig` command to enable and disable services when the system boots, and the `service` command to start and stop services that are currently running. First:

1. Disable NetworkManager at boot time:
`sudo chkconfig NetworkManager off`
2. Enable the network service at boot time: `sudo chkconfig network on`
3. Stop the NetworkManager right now: `sudo service NetworkManager stop`
4. Start the network service right now: `sudo service network start`

Next **configure the static IP address for our server**. To do that, we have to create a file that will tell the network service what parameters to use to configure the network interface. In Linux, NICs are called `ethX`, where X is a number that usually starts with 0. Since we only have one interface on our VM, it is called `eth0`. So, we need to write a configuration file with all the information needed to setup `eth0` at boot time.

In Centos , these interface-specific configuration files live in `/etc/sysconfig/network-scripts/`. Inside that directory, you will write a file named `"ifcfg-eth0"` that contains the configuration for the interface. Each line is of the form `PARAM=VALUE` and is used to set one specific piece of information.

Before we can build the file, we need to know what our networking configuration is actually going to be. We will use the `192.168.10.0/24` network for our VMware network. Specifically, we will assign our server an address of `192.168.10.10` (skipping the first few addresses because VMware likes to use them for its own purposes). The gateway router is `192.168.10.2` (again, from VMware). That is all we need for now, so do the following:

1. Open up a terminal.
2. Go to the `/etc/sysconfig/network-scripts` directory.
3. Backup `ifcfg-eth0` but give a completely new name .
4. Edit `ifcfg-eth0` with your editor of choice (and `sudo`).
5. Add the following lines:
 - a. `DEVICE=eth0`
 - b. `BOOTPROTO=static`
 - c. `ONBOOT=yes`
 - d. `IPADDR=192.168.10.10`
 - e. `NETMASK=255.255.255.0`
 - f. `GATEWAY=192.168.10.1`
6. Save the file and exit your editor. Note some VMs prefer the gateway to be `192.168.10.2`

7. For the change to take effect, you have to **restart the network** service: `sudo service network restart`
8. To **see if it actually worked**, use the `ifconfig` command to show the current status of `eth0`: `ifconfig eth0`
9. You should see that your IP address (`inet addr`) is 192.168.10.10.
10. **TAKE A SCREENSHOT** of the output of `ifconfig`, showing the command you ran.

The next step is to **make sure that DNS is still configured** to use the VMware DNS server. The DNS configuration file is `/etc/resolv.conf`, and the DNS server in VMware is 192.168.10.1 (the same as the gateway router). So:

1. Check `resolv.conf`: `cat /etc/resolv.conf`
2. Look for a line that starts with "nameserver", and it should say "nameserver 192.168.10.1".
3. If there is a different IP address for the nameserver, edit the file and change it to be 192.168.10.1.
4. If there is no nameserver line, then you need to one:
 - a. `nameserver 192.168.10.2`

It's always a good idea (when possible) to **make sure that everything works as expected after a reboot** when you change services and whatnot that start at boot time. Let's do that next:

1. Reboot your VM (command line, GUI, whatever).
2. When it reboots, log back in and open up a terminal.
3. Run `ifconfig eth0` again to make sure the IP address is correct (192.168.10.10).
4. Try pinging out to make sure you have connectivity: `ping www.google.com`
5. Do a CTRL-C to stop the ping after a few get through.
6. **TAKE A SCREENSHOT** of the output of the ping command showing that a few pings successfully got back.
7. Of course, if it didn't work, go back through the steps until you get everything configured correctly.

STEP 5: We are ready to install and configuring our DHCP server. First install the DHCP server package, simply called `dhcp`:

1. Run: `sudo yum install dhcp`

The default location for the DHCP configuration file is `/etc/dhcp/dhcpd.conf`. There is a sample file with some different options and examples in `/usr/share/doc/dhcp-3.0.5/dhcpd.conf.sample`, but there is a bunch of stuff in

there that we don't want, so it's easier to start from scratch in this case. You should still look at the sample file to get an idea of how the file looks.

There are two required parts to the configuration file: global options and subnet definitions. Global options affect every subnet that the DHCP server is managing. There are few things that you usually set server-wide, and any common DHCP options that you share across subnets could go there as well. Most of the time, admins opt to put more into the individual subnet definitions even if it is a bit redundant just so that they see all the configuration information for one subnet in the same place. We are only going to set two global options that are generally good to set (or not) explicitly:

1. Edit `/etc/dhcp/dhcpd.conf` with your editor of choice (with `sudo`).
 - a. If you are blocked from editing (or even seeing) the file, login as root and do the edits.
2. **Add** the following two lines:
 - a. `ddns-update-style none;`
 - b. `authoritative;`
3. Note that each line ends with a semi-colon, as will most lines in the file.
4. The first option says that we are not going to try to automatically synchronize DNS state with DHCP state.
5. The second says that this is the authoritative DHCP server for the LAN.
6. Look at the man page for `dhcpd.conf` to read more about these options.
7. Leave your editor open, as we have more work to do.

Next up, we need to specify a subnet section that tells DHCP what IP addresses it can hand out to clients, and what other DHCP options should go along with it. The subnet section starts with a line that tells DHCP what the network prefix and netmask are, and looks like "`subnet NETWORK_PREFIX netmask SUBNET_MASK`". For our network, recall that we are using 192.168.10.0/24, so the network prefix is 192.168.10.0 and the netmask is 255.255.255.0. Then, you add a pair of curly braces (like after an `if` statement in C), and any options that are part of this subnet go in between those braces. So, do:

1. **ADD** the following three lines to your `dhcpd.conf`:
 - a. `subnet 192.168.10.0 netmask 255.255.255.0`
 - b. `{`
 - c. `}`
2. Note there is NOT a semi-colon after the subnet line because it is followed by the curly braces on the next line (again, like `if`-statements in Perl).

Now we need to set the parameters and DHCP options for the subnet. There are 5 things that you nearly always set: the default and max lease times, router information, and DNS servers and domain information. The lease times tell DHCP how long it can lease out addresses to clients

and should be configured according to your site's needs. We will set a default lease time of 1 hour and a maximum lease time of 1 hour. The other three values are DHCP options that get sent with every DHCP lease. The router and DNS server are the VMware built-ins for now: 192.168.10.2. The DNS domain name is "comp150" for us, but in general is the domain name of your organization (e.g., "wit.edu"). Let's add that stuff in:

1. **Add** these lines in between the two curly braces.
2. Add these two lines:
 - a. `default-lease-time 3600;`
 - b. `max-lease-time 3600;`
3. These parameters set the lease times used by the server, in seconds.
4. Then **add** these three lines:
 - a. `option routers 192.168.10.2;`
 - b. `option domain-name "comp150";`
 - c. `option domain-name-servers 192.168.10.2;`
5. Again, these values are actual DHCP options that will be sent to clients in this subnet.

Finally, we need to tell DHCP what the range of addresses is that it is actually allowed to give to dynamic clients. To do that, use the "range" option. Without this, DHCP will assume that it can't give out addresses to clients (unless they are listed explicitly in the subnet). So, do:

1. **Add** one more line (again, in between the curly braces):
 - a. `range 192.168.10.20 192.168.10.250;`
2. This tells DHCP that it can freely give out addresses between 192.168.10.20 and 192.168.10.250.
3. We are setting aside the addresses from 192.168.10.1 through 192.168.10.19 for servers and other static assignments.
4. Save your work and exit.

Any time you update the DHCP configuration file, you need to run a check to make sure that there are no syntax errors or typos. You can do that with the `dhcpcd` program and the `-t` option:

1. **Run:** `sudo dhcpcd -t`
2. If you have any errors you will see output that tells you to look at certain lines. As usual, look at the line given and the line above.
3. Keep fixing your errors until there are none listed in the `dhcpcd -t`.

Alright, now it's time to give it a go. First, we need to set up DHCP to run at boot time, then start up the service right now:

1. Set the `dhcpcd` service to run at boot time: `sudo chkconfig dhcpcd on`
2. Now actually start the `dhcpcd` service: `sudo service dhcpcd start`

3. If you did everything right, you will get the OK message saying that it started. If you get the FAIL message then you've probably got an error in your `dhcpd.conf` file. Go back through the process above to find and fix your errors.

STEP 6: The DHCP server is now running, so we need a client to test it out. Let's start with the **centos-client VM**:

1. Power on your centos-client VM. (If you suspended it, you must reboot to get your new ip address.)
2. Log in.
3. Open up a terminal.
4. Run: `ifconfig`
5. You should see an IP address assigned to `eth0` in the range between 192.168.10.20 and 192.168.10.250.
6. If not, you made a mistake somewhere which you'll have to fix.
7. To further verify that we are actually talking to our DHCP server, we need to look at the `dhclient` lease file:
 - a. Change directories to `/var/lib/dhclient`
 - b. List the files in reverse (`-r`) time (`-t`) order so that the bottom file in the output is the most recently used: `ls -ltr`
 - c. `cat` the file you identified in the above command to see the lease information.
8. You should see a lease that says that the DHCP server (`dhcp-server-identifier`) is 192.168.10.10, and the DNS server and router is 192.168.10.2.
9. **TAKE A SCREENSHOT** of the output of the `cat` command showing the lease from your DHCP server.

Now, do the same for **the windows7-client** (or windows8-client) VM:

1. Power on your windows7-client VM.
2. Log in.
3. Open up a cmd prompt.
4. Run: `ipconfig /all`
5. Find the section for your Local Area Connection.
6. You should see an IP address of 192.168.10.21, a DHCP server of 192.168.10.10, and a DNS server and gateway router of 192.168.10.2.
(You might need to `/release` and `/renew` to get a new ip address.)
7. **TAKE A SCREENSHOT** of the output of the `ipconfig /all` command showing the correct networking information.
8. You can shut down or pause this VM now if you want to save resources on your laptop.

Now that we've got the basics working, let's also **add a reservation for a specific IP address for the centos-client VM**. Before we start editing the DHCP configuration file again, we need to know the MAC/Ethernet address of the client because that is how the DHCP server will uniquely identify the client. So, on the centos-client VM:

1. Open a terminal and run: `ifconfig`
2. You should see an entry for `eth0`.
3. Look for the Ethernet address (`HWaddr`) and copy/write it down.

Back on the centos-server VM, we need to add a special entry for the centos-client VM in the `dhcpd.conf` file. We don't want to assign it an address inside the dynamic range we configured earlier (those are for dynamic clients). Instead, we will give it a fixed IP address of 192.168.10.15. So, on the centos-server VM:

1. Edit `/etc/dhcp/dhcpd.conf`.
2. Go to the bottom of the file, after the subnet entry (i.e., after the closing curly brace for the subnet).
3. We will add an entry to tell DHCP to give the centos-client VM a fixed address by adding a host entry.
4. First **add** the following three lines:
 - a. `host centos-client`
 - b. `{`
 - c. `}`
5. That defines a host entry for a host named centos-client.
6. Now we need to add two lines that tell DHCP how to identify the client and what address to give it.
7. **Add** the following two lines in between the host curly braces:
 - a. `hardware ethernet MAC_ADDRESS;`
 - b. `fixed-address 192.168.10.15;`
8. Obviously, replace `MAC_ADDRESS` with the actual MAC address of your centos-client VM.
9. Save the file and exit.
10. Always double check for syntax errors after editing `dhcpd.conf` by running:
`sudo dhcpd -t`
11. Fix any errors listed.
12. Now restart the `dhcpd` service so that it re-reads the configuration file:
`sudo service dhcpd restart`

To test everything out, head back to your centos-client VM. We need to do the Windows equivalent of an `ipconfig /release` and `ipconfig /renew` to get the new IP address

from DHCP. Use the `dhclient -r` to release and `dhclient` to renew. When you execute the latter you will actually see the dhcp negotiation. On your centos-client VM:

1. Open a terminal.
2. `sudo /sbin/dhclient -r`
3. run `ifconfig` to be sure the ip address was released.
4. `sudo /sbin/dhclient`
5. **TAKE A SCREENSHOT** showing both commands.