

Lab Assignment 2

Linux System Lab

1. Lab Overview

All labs this semester will require each student to have their own Linux VM servers on Jetstream2 so that they can modify/break without affecting other students. The objective of this lab is to set up multiple server services to use throughout the semester. All of the work in this lab will be performed through the command line. You may either SSH into your machine or use Jetstream's web desktop. All the assignments below should be done in both VMs (i520-client and i520-server). Use SSH to accomplish all labs.

Do not change anything on your Jetstream2 host machine.

2. VM snapshot

A snapshot is basically a “frozen picture” of the disk, the memory and the state of the devices associated with a domain, taken in a specific moment in time. Snapshots are very useful: they can be created, for example, before performing a potentially dangerous task on the system, in order to have an easy way to switch back to a well-known state.

We will perform operations to change the configurations of the VMs and it is better to create a snapshot beforehand.

All the operations below are run from your Jetstream host.

First, we need to convert the disk type from raw to qcow2, make sure both the vms are shutdown and run the following commands:

```
sudo qemu-img convert -f raw -O qcow2 /var/lib/libvirt/images/server.raw  
/var/lib/libvirt/images/server.qcow2
```

```
sudo qemu-img convert -f raw -O qcow2 /var/lib/libvirt/images/client.raw  
/var/lib/libvirt/images/client.qcow2
```

And then, reconfig the server:

```
virsh edit Server-ubuntu-20.04
```

Choose an editor and change the lines from:

```
<driver name='qemu' type='raw'/>  
<source file='/var/lib/libvirt/images/server.raw'/>
```

to

```
<driver name='qemu' type='qcow2'/>  
<source file='/var/lib/libvirt/images/server.qcow2'/>
```

Similarly for the client:

```
virsh edit Client-ubuntu-20.04
```

Change

```
<driver name='qemu' type='raw' />
<source file='/var/lib/libvirt/images/client.raw' />
```

to

```
<driver name='qemu' type='qcow2' />
<source file='/var/lib/libvirt/images/client.qcow2' />
```

You can create/revert/delete a snapshot with the following commands:

```
virsh snapshot-create-as --domain {VM-NAME} --name {SNAPSHOT-NAME}
```

```
virsh snapshot-revert --domain {VM-NAME} --snapshotname {SNAPSHOT-NAME} --running
```

```
virsh snapshot-delete --domain {VM-NAME} --snapshotname {SNAPSHOT-NAME}
```

Replace {VM-NAME} with the name of VM, i.e., Server-ubuntu-20.04 or Client-ubuntu-20.04, and {SNAPSHOT-NAME} with a name for the snapshot.

3. Network configuration

3.1 Play with network configuration

Add a name server “8.8.8.8” to /etc/resolv.conf. If you manually update /etc/resolv.conf, your change would not stay permanently. Determine how Ubuntu updates /etc/resolv.conf and make your changes persistent.

Please keep the existing name server (e.g. nameserver 129.79.1.1 as a primary name server).

3.2 Becoming Familiar with the Aptitude Package Manager

Software packages can be installed/updated from the Ubuntu repositories via the Aptitude package manager. In order to complete the various objectives in this lab, you will need to install some software packages using this tool. There is an introduction and links to more resources here: <https://manpages.ubuntu.com/manpages/xenial/man8/apt.8.html>

To verify that the software list cache on your server is up-to-date, run **sudo apt update**. This will have the server download the latest information about packages available. This command also needs to be run whenever repositories are added or removed from the system.

Now, to update all of the software on the system and install any necessary dependencies, run **sudo apt dist-upgrade** .

To install a new package, run **sudo apt install <package name>**. For example, run **sudo apt install fping**. Once the installation is complete, you should be able to run **fping example.org** and it should return “**example.org is alive**”.

4. Lab Tasks

4.1 Install Apache2

Use the aptitude package manager to install the apache2 web server. Installation instructions and basic configuration tasks can be found here:

<https://ubuntu.com/server/docs/web-servers-apache>

For now, you will not need to change any of the default installation configurations.

If the installation was successful, you should be able to type your server's IP address into a web browser and get a web page back that says “It Works!”

4.2 Install MySQL Server

Install the “mysql-server” package. During the installation you need to set the password for the MySQL root user to **s3cur3n3tworks#!@hac** which is used to manage the databases and users. Installation and basic configuration instructions can be found here:

<https://ubuntu.com/server/docs/databases-mysql>

To test the installation, you should be able to run “**mysql -u root -p**” and then enter the root password when prompted. If the installation was successful, you should get a prompt that says ‘mysql>’. Type ‘quit’ to exit.

4.3 Install PHP and the PHP Apache modules

Install the php, php-cli, php-mysql, and libapache2-mod-php packages. You will not need to make any configuration changes, but you will need to restart apache2 for the new module to take effect. Restart apache2 with the following command: **sudo service apache2 restart**

To test it, we need to put a page in the web server document root that uses PHP. Apache's default document root is ‘/var/www/html’.

So run **sudo nano /var/www/html/group.php** and enter the following into the file:

```
<?php
echo "Hello from class: i520/B544";
?>
```

Then press CTRL-X to save and exit the editor. If the installation worked, you should be able to visit **http://<your server's IP address>/group.php** and see the text “*Hello from class: i520/B544*” without the quotes.

4.5 Add a Partially Privileged User (sudoers)

Any of the commands that made changes to the system required **sudo** to be prepended to them. This command allows a user to execute a program as another user (by default root). The administrative account being used is granted access to do this by being a member of the *admin* group which has an entry in the *sudoers* file. The *sudoers* file can be used to allow users to only run certain programs with root privileges. Using the man pages for sudoers and other online materials, complete the following tasks on both servers:

- Use “**adduser**” to create a new user with the username **serveroperator** and the password **1powerS3rv3rs**
- Use the “**visudo**” command to add an entry to the sudoers file that permits the *serveroperator* user to run the following commands: **service**, and **reboot** .

You can test this by logging in as the *serveroperator* user and running **sudo reboot** , which should reboot your server; and, **sudo service apache2 restart** , which should restart apache2. The user should not be able to run any other commands with sudo. (e.g **sudo ls** should generate an error).

Evaluation

Completion of the lab will be evaluated by the AIs, by logging in to your VMs and checking the following

1. Add a new DNS server to */etc/resolv.conf* permanently.
2. Proper setup of MySQL server
3. Proper setup of PHP
4. Setup partially privileged user at */etc/sudoers*