

# Skills Practice - Managing Software

## **Goal:**

This goal of these practice exercises is to familiarize you with managing software with the `dnf` command. You will be using these skills in future projects in this course.

## **Instructions:**

### **Start a Virtual Machine**

For this practice we can use a virtual machine that we created in a previous project. First, start a command line session. Change into your `linuxclass` folder and then change into the `testbox01` directory.

```
cd linuxclass
cd testbox01
```

Next, start the virtual machine using the `vagrant up` command. If the virtual machine is already running, `vagrant` will let you know that it's ready to use. If it's stopped or paused, `vagrant` will start the virtual machine.

Start the virtual machine and connect to it.

```
vagrant up
vagrant ssh
```

### **Determine Your Username**

You can probably tell what user you are by looking at the command prompt, but make sure.

```
whoami
```

The command should have reported that you are indeed logged into the Linux system as the `vagrant` user.

## Determine What Sudo Commands You Can Run

To install software you need superuser (root) privileges. Make sure you can run the dnf command as root.

```
sudo -l
```

The very last line of that output should look very similar to this: (ALL) NOPASSWD: ALL

That means you can run any command without a password as any user.

## Find the Apache Web Server Package

You don't need superuser privileges to search for software, but you do need them to install or remove software. Search for apache.

```
dnf search apache
```

You'll get a very long list of packages. Let's narrow the search by adding another term, or keyword. This time search for apache and server.

```
dnf search apache server
```

This shorter list narrows down the possible choices. Based on the short descriptions you can probably guess that the httpd package contains the apache web server software. To make sure, get more information about this package.

```
dnf info httpd
```

Notice the information that is displayed. It tells you the version of the software, the repository from which it will get installed, and gives you a more detailed description.

## Install the Apache Web Server

Now that you've found the package for the Apache web server, install it. Remember that installing software requires superuser privileges. Typically, you will log into a Linux system as a normal user and then use sudo when you need superuser privileges.

Install Apache with dnf, using sudo.

```
sudo dnf install httpd
```

Dnf will run a dependency check and report back to you what packages it will install to meet the dependencies. Some packages don't have dependencies and sometimes all the dependencies will already be installed. In any case, dnf will ask you to confirm the installation. Type "y" and hit enter to allow dnf to perform the installation.

By the way, if you forget to run the "dnf install" command with superuser privileges, dnf will remind you to do so. Try it out:

```
dnf install httpd
```

Dnf will politely tell you that you need to be root to perform the installation.

## Start the Apache Web Server

Now you are ready to start the web server. Starting and stopping system wide services requires superuser privileges. Use sudo to start httpd.

```
sudo systemctl start httpd
```

Open up a web browser on your local machine and type in the IP address of your Linux system: 10.23.45.10. If Apache started, you will see a test page.

We want to make sure that the web server starts if or when we reboot the system. Do that by enabling it with systemctl.

```
sudo systemctl enable httpd
```

You can combine both steps of starting and enabling a service by using the --now option to the enable subcommand of systemctl:

```
sudo systemctl enable --now httpd
```

## Install and start MariaDB

Search for MariaDB.

```
dnf search mariadb
```

Look at the mariadb package. Can you tell what this package does?

```
dnf info mariadb
```

In the description of the mariadb package it states that it contains the MariaDB/MySQL client programs. We'll need the client to connect to the database server, but we also need the server! Look at the mariadb-server package.

```
dnf info mariadb-server
```

That package contains the server. Let's install it. Of course, you need superuser privileges to install software. We've already done that with sudo. Let's first switch to the root user and confirm we are indeed root.

```
sudo -s  
whoami
```

Now that we are the superuser (root), we no longer need to prepend sudo to commands that require root privileges. Install the mariadb-server package as root.

```
dnf install mariadb-server
```

Confirm the installation by typing "y" and hitting enter.

Start the database server and enable it so it starts on boot. We're already root, so sudo is not required.

```
systemctl start mariadb  
systemctl enable mariadb
```

Run the mysqlshow command to confirm.

```
mysqlshow
```

## Add the EPEL repository

Let's install a little utility called htop. It is used to view process and system performance information. We won't really be using the htop utility in the course, but we can use it as an excuse to practice using EPEL. First, look for the htop package.

```
dnf search htop
```

Dnf will not find any matches because this package is not in the standard repository. Let's add the EPEL repository to the system. Since we're root, we don't need to use sudo. If you're getting tired of confirming all the installations that dnf is performing you can use the "-y" option to automatically answer yes and trust dnf. Let's do that here.

```
dnf install -y epel-release
```

Now, search for htop again. It should return a result.

```
dnf search htop
```

View the information about this package. Notice what repository the package is coming from.

```
dnf info htop
```

Install it:

```
dnf install -y htop
```

You can run the command to test it. Hit "q" to quit.

```
htop
```

By the way, the htop utility aims to improve the top command which is included by default on Linux.

## Uninstall a package

Let's assume you no longer need htop. Remove it.

```
dnf remove htop
```

After you confirm the removal by typing "y", the package will be removed. Note: you can automatically answer yes while removing software by using the "-y" option. (`dnf remove -y htop`)

Confirm the command is gone by trying to execute it:

```
htop
```

## Install PHP

Let's install the final component of the LAMP stack: PHP. First, let's return to our normal user account by typing `exit`:

```
exit
```

Now we need to use `sudo` for superuser privileges. I happen to know that PHP is installed with the `php` package. Also, the MySQL PHP module is made available through the `php-mysqlnd` package. Let's install them both now.

```
sudo dnf install -y php php-mysqlnd
```

When you update configurations related to a service, you should reload the service. The same is true if you are installing or removing software that interacts with a service as is the case with PHP and Apache.

```
sudo systemctl reload httpd
```

Note that "`systemctl reload`" tells the service to re-read its configuration files. Not all services can do that and not all services honor the reload command. If you want to make sure the configuration takes effect, use "`systemctl restart`". Said another way, if `reload` doesn't work, use `restart`.

```
sudo systemctl restart httpd
```

You've now installed all the major components of the LAMP stack!