Return to Syllabus (/cp/modules/view/id/33)

Exercise: Exercise: Installation and Image Setup

1 2

1. Update your system as appropriate for the distribution you are using. Once updated, for the duration of this exercise, completely disable SELinux on the system and reboot. Once logged in, verify that SELinux is disabled.

```
 [user@linuxacademy:~] sudo yum update (or upgrade)

 (Output) Repo updates and packages needing to be updated
will be listed here

 [user@linuxacademy:~] sudo vim /etc/selinux/config

 (Output)...

 Find the line: SELINUX=enforcing

 Set it to: SELINUX=disabled

 Save the file and exit

 (Output)...

 [user@linuxacademy:~] cat /etc/selinux/config | grep
"SELINUX="
```

2. Using the appropriate package management commands, install the 'docker' package. Once installed, enable the service so that it starts upon reboot. Additionally, start the 'docker' service and verify it is running.

```
[user@linuxacademy:~] sudo yum -y install docker

(Output) Package downloads and installs here

CentOS 6.x SOLUTION

[user@linuxacademy:~] sudo chkconfig docker

[user@linuxacademy:~] sudo service docker start

CentOS 7.x SOLUTION

[user@linuxacademy:~] sudo systemctl enable docker

[user@linuxacademy:~] sudo systemctl start docker

[user@linuxacademy:~] ps aux | grep docker

(Output)

root      5802  0.0  0.3 346424 12936 ?        Ssl
 21:02   0:00 /usr/bin/docker -d --selinux-enabled

(Output)
```

3. Enable the non-root users to run 'docker' commands on the system. Create the appropriate group, add the users you want to have access to 'docker' commands, restart the 'docker' service and verify that non-root users can run basic 'docker' commands.

```
[user@linuxacademy:~] sudo groupadd docker

[user@linuxacademy:~] vim /etc/group

(Output)...

Find the line that looks like: docker:x:1001:

Add the 'user' user to the end of that line (after the
:)

(Output)...

[user@linuxacademy:~] sudo service docker restart (OR
sudo systemctl restart docker)

[user@linuxacademy:~] docker images

(Output)

REPOSITORY          TAG                 IMAGE ID
    CREATED             VIRTUAL SIZE

(Output)
```

4. Once 'docker' is installed and non-root users can run commands, use the appropriate 'docker' commands and options to download the latest available image in the public repository for Ubuntu. Once downloaded and installed, verify the image appears in the local base image list.

```
[user@linuxacademy:~] docker pull ubuntu:latest

(Output) NOTE: Your output may differ if image has been
updated

latest: Pulling from docker.io/ubuntu

6071b4945dcf: Pull complete

5bff21ba5409: Pull complete

e5855facec0b: Pull complete

8251da35e7a7: Pull complete

Digest:
sha256:1572e29178048ad9ab72e78edd4decc91a3d8a8dea0ca39817efc7c

Status: Downloaded newer image for
docker.io/ubuntu:latest

(Output)...

[user@linuxacademy:~] docker images

(Output) NOTE: Again, your output may differ slightly

REPOSITORY          TAG                 IMAGE ID
   CREATED             VIRTUAL SIZE

docker.io/ubuntu    latest              8251da35e7a7
   11 days ago         188.3 MB

(Output)...
```

5. Start a container based upon the Ubuntu base image downloaded in Step #4. When you start the container, start it connected to the current terminal, in interactive mode, starting the bash shell for you to connect to. You may exit the container at that time.

```
[user@linuxacademy:~] docker run -it
docker.io/ubuntu:latest /bin/bash

(Output) NOTE: Your output will differ based on the
container ID assigned

root@7aaf3de3ed4f:/# exit
```

Back Completed