**Exercise: Docker Installation and Image Setup**

***NOTE:***These Docker exercises can be completed using the Linux Academy Lab Servers running CentOS 7. Although the solutions are virtually identical on Debian/Ubuntu systems, the package management commands and service management commands will differ. NOTE: You must use the Docker repository as defined on docs.docker.com for your chosen distribution and version and you MUST install version 1.10+ in order for all exercises to complete as indicated.

1. Update your system as appropriate for the distribution you are using. Use the instructions in the videos OR on the Docker site to add the DOCKER repository for installing the latest copy of Docker for your distribution and version.

[user@minion-01:~] sudo yum update (or upgrade)

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@minion-01:~] sudo yum -y install docker

(Output) Package downloads and installs here

CentOS 7.x SOLUTION

[user@minion-01:~] sudo systemctl enable docker

[user@minion-01:~] sudo systemctl start docker

[user@minion-01:~] 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@minion-01:~] sudo groupadd docker

[user@minion-01:~] 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@minion-01:~] sudo service docker restart (OR sudo systemctl restart docker)

[user@minion-01:~] 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@minion-01:~] 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: sha256:1572e29178048ad9ab72e78edd4decc91a3d8a8dea0ca39817efc7cf2d86c6d7

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

(Output)...

[user@minion-01:~] 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@minion-01:~] docker run -it docker.io/ubuntu:latest /bin/bash

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

root@7aaf3de3ed4f:/# exit