Lab Experiment-9

- **Build Docker Image using .dockerignore file.**

There are cases when we have some sensitive data like our passwords files or some other unwanted data such as some large files that would just make the image bulky. While creating image, we need to ignore such files from copying.

This task can be achieved by using dockerignore file. All we need to do is to add the file names to be ignored in the dockerignore file and then these files will not be copied.

Here, we will see how it works. The steps that need to be followed are:

1. Create a file named "hello.txt". This is the file that we need to have copied in our image.

Command: touch hello.txt

```
vagrant@ubuntu-xenial:~/ubuntuImage$ touch hello.txt
```

2. Create a couple of miscellaneous files. These are the files that you want to ignore while making the image. These files are named "password.txt" and "bigfile.png". Here it is being assumed that the first file has some sensitive passwords and the second file is a large sized unwanted image that will just make the image bulky.

Command: touch password.txt bigfile.png

```
vagrant@ubuntu-xenial:~/ubuntuImage$ touch password.txt bigfile.png
```

The contents of the directory after execution of the above steps are as follows:

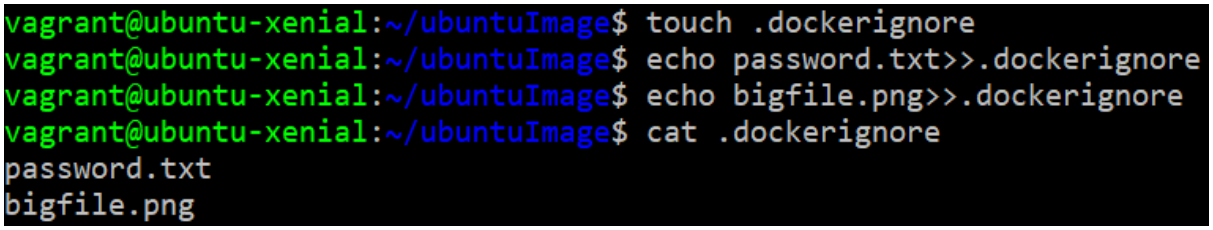
```
vagrant@ubuntu-xenial:~/ubuntuImage$ ls
bigfile.png  hello.txt  password.txt
```

3. Make a Dockerfile and add some code in it as shown below.

Command: nano Dockerfile

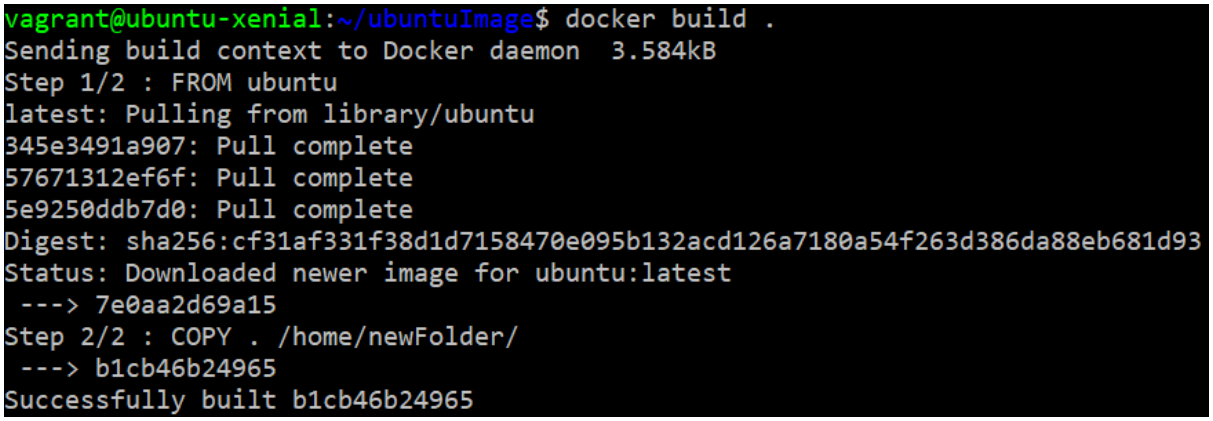
This file will create an image of Ubuntu and will have the contents of the current directory copied inside home/newFolder directory.

4. Create a .dockerignore file in the directory and add the filenames that you want to ignore.



5. Now we are ready to build the image.

Command: docker build .



6. Now we will run a container of this image. To do so, we can use the image id.

Command: docker images

```
vagrant@ubuntu-xenial:~/ubuntuImage$ docker images
REPOSITORY    TAG        IMAGE ID       CREATED              SIZE
<none>        <none>     b1cb46b24965   About a minute ago   72.7MB
ubuntu        latest     7e0aa2d69a15   11 days ago          72.7MB
```

As we can see, the id of our image is the one starting with b1cb. Run the container using this image id.

Command: docker run –it –d <image-id>

```
vagrant@ubuntu-xenial:~/ubuntuImage$ docker run -it -d b1
a6064e22e000cb8ab0591650d907c49188305c8822adee7224fa3ac267fdbd9d
vagrant@ubuntu-xenial:~/ubuntuImage$ docker ps
CONTAINER ID   IMAGE     COMMAND       CREATED         STATUS         PORTS     NAMES
a6064e22e000   b1        "/bin/bash"   3 seconds ago   Up 1 second              zen_lovelace
```

7. Open the terminal of the container and check the contents of the /home/newFolder directory.

Command: docker exec –i –t <container-id> /bin/bash

```
vagrant@ubuntu-xenial:~/ubuntuImage$ docker exec -i -t a6 /bin/bash
root@a6064e22e000:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp
root@a6064e22e000:/# cd home
root@a6064e22e000:/home# ls
newFolder
root@a6064e22e000:/home# cd newFolder
root@a6064e22e000:/home/newFolder# ls -ah
.  ..  .dockerignore  Dockerfile  hello.txt
root@a6064e22e000:/home/newFolder#
```

As you can see, the unwanted files are ignored.