Docker Assesment:

* Create a Linux Virtual server in (EC2 Instance) AWS with name “ipsraws\_demo”.
* Connect EC2 server using Git Bash from your local machine.
  + Download AWS keypair and save it into local machine
  + Install Git in your local machine.
  + Connect with EC2 server using ssh client from Git Bash.
    - e.g.: ssh -i "DevOps.pem" [ubuntu@ec2-3-94-210-89.compute-1.amazonaws.com](mailto:ubuntu@ec2-3-94-210-89.compute-1.amazonaws.com)
* Create a predefined ubuntu docker image from Dockerd Hub using pull.
  + docker pull ubuntu
* List all docker images
  + docker images
* Convert ubuntu docker image as file and save to local drive
  + docker save ubuntu -o ubuntu.tar
* Delete the ubuntu docker image
  + docker rmi -f ubuntu
* Convert ubuntu docker file as docker image.
  + Docker load -i ubuntu.tar
* Start a new container with name ‘MyUbuntuContainer’ from created ubuntu image in interactive mode.
  + docker run --name myubuntucontainer -it ubuntu bash
* Check Apach2 service is installed in ‘MyUbuntuContainer’ container.
  + Service apache2 status
* Install Apache2 service in ‘myubuntucontainer’ if apache2 service is unrecognized and up the service
  + apt-get update
  + apt-get install -y apache2
  + Make apache2 service up and running.
* List all the running containers from your AWS Virtual server
  + docker container ls
* List all the process created/running in your virtual instance
  + docker ps -a
* Login into dockerhub from the terminal (EC2 Instance).
  + docker login
    - provide your dockerhub user name and password (During first time login in the same VM)
  + Create a copy or tag of custom image (MyUbuntuImage) to push into dockerhub
    - docker tag myapacheimage1: latest <dockerhub name>/ MyUbuntuImage

e.g: docker tag myapacheimage1: latest ipsraws/ MyUbuntuImage

* + docker push <tagged custom image>
    - docker push ipsraws/MyUbuntuImage
    - Check Docker hub whether ‘ipsraws/MyUbuntuImage’ is pushed into Git Hub server
* Create a custom image (Instead of pulling predefined image from Git Hub) of simple application using the following steps:
  + Login into EC2 server using the terminal
  + Create the text file without .txt extension named Dockerfile (Buildscript)
  + Demo : Create custom Image to install Apache Web Server
    - Below is the sample script for Dockerfile
      * + From Ubuntu
        + MAINTAINER <your name>
        + RUN apt-get update
        + RUN apt-get install -y apache2
        + RUN apt-get clean
        + Run -rf /var/lib/apt/lists/\*
        + ENV APACHE\_RUN\_USER www-data
        + ENV APACHE\_GROUP www-data
        + ENV APACHE\_LOG\_DIR /var/log/apache2
        + EXPOSE 80
        + CMD [“/usr/sbin/apache2”, “-D”, “FOREGROUND”]
      * Build the image called ‘MyApacheCustomImage’ from the above mentioned Dockerfile script by running the below command
        + docker build -t <custom image name> <.>

“.” is the present directory where Dockerfile created.

e.g., docker build -t MyApacheCustomImage’.

* + - * + Check Apach2 service is installed in your Virtual machine

service Apache2 status – If it is unrecognized, get Apache2 installed.

* + - * Create the container called MyApacheContainer from the above created custom image
        + docker run -p 8080:80 --name= MyApacheContainer MyApacheCustomImage’

Here -p is to specify the port. 8080:80 –

**8080(host):80 (Container)** – Apache is running on port number 80 in the container which is mapped to the port number 8080 of the host machine (EC2 server).

Get the public IP address of the virtual machine using either one of the below commands

Ifconfig

curl wgetip.com:

Run the below URL from your local machine

http://<public IP of EC2 instance>:8080/

e.g. : <http://34.230.58.153:8080/>

* Create your own webpage name <yourname> with html extension
  + Go to /var/www/html
  + Create a file with below content <yourname>.html

http://<public IP of EC2 instance>:8080/<yourname>.html

* + - * Destroy the container
        + docker container prune