Setting up a Docker instance

Type the following command to check the docker version installed on lab:

docker version

```
root@ip-172-31-17-73:~# apt install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
docker.io is already the newest version (18.09.7-0ubuntu1~18.04.3).
The following packages were automatically installed and are no longer required:
  apache2-bin apache2-data libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0
Use 'apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 9 not upgraded.
root@ip-172-31-17-73:~# docker version
Client:
 Version:           18.09.7
 API version:       1.39
 Go version:        go1.10.1
 Git commit:        2d0083d
 Built:             Wed Jul  3 12:13:59 2019
 OS/Arch:           linux/amd64
 Experimental:      false

Server:
 Engine:
  Version:          18.09.7
  API version:      1.39 (minimum version 1.12)
  Go version:       go1.10.1
  Git commit:       2d0083d
  Built:            Mon Jul  1 19:31:12 2019
  OS/Arch:          linux/amd64
  Experimental:     false
root@ip-172-31-17-73:~#
```

Building a custom Docker image to be deployed

- First, clone the Git repository on Docker host using the command below:

  git clone https://github.com /Docker.git

- Run with docker build command to build a custom Docker image

  cd Docker

docker build -t phpcode . -f Dockerfile

```
root@docker:~/Docker# docker build -t phpcode .
Sending build context to Docker daemon  337.9kB
Step 1/14 : FROM ubuntu
 ---> 93fd78260bd1
Step 2/14 : ENV DEBIAN_FRONTEND=non-interactive
 ---> Using cache
 ---> b21eb69f632a
Step 3/14 : RUN apt-get update -y
 ---> Using cache
 ---> d2e4866734b9
Step 4/14 : RUN apt-get install -y git curl apache2 php libapache2-mod-php php-mysql
 ---> Using cache
 ---> 85f084edfc0b
Step 5/14 : RUN rm -rf /var/www/html/*
 ---> Using cache
 ---> b56166da0f16
Step 6/14 : ADD src /var/www/html/
 ---> Using cache
 ---> ba9e5c5c651c
Step 7/14 : RUN a2enmod rewrite
 ---> Using cache
 ---> cff3e4bb8c42
Step 8/14 : RUN chown -R www-data:www-data /var/www/html
 ---> Using cache
 ---> 7a4314c7b69b
Step 9/14 : ENV APACHE_RUN_DIR /var/www/html
 ---> Using cache
 ---> 663a68663f90
```

- Once the image is built, check if it is built properly or not. You can see a Docker image entry using Docker images command

```
Removing intermediate container 66720df3cf7e
 ---> b914fd976a06
Successfully built b914fd976a06
Successfully tagged phpcode:latest
root@ip-172-31-17-73:~/Docker# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
phpcode             latest              b914fd976a06        3 minutes ago       251MB
ubuntu              latest              4c108a37151f        2 weeks ago         64.2MB
root@ip-172-31-17-73:~/Docker#
```

Initializing a Docker swarm cluster and deploying a container to the cluster

- First, we need to initialize Docker swarm using the set of commands given below:

docker swarm init

docker node ls

- Once the node is configured, deploy the custom Docker image on the Docker swarm cluster following the process shown below

docker service create -p 80:80 --name webserver phpcode

docker service ls

curl localhost