**Assignments**

**Docker Questions:**

1) Spin up a temporary container with image nginx:1.19.10 and execute inside it, such that the container should be destroyed, once you exit from the container

2) Spin up a container with image nginx:1.19.10 such that it should restart automatically if any fatal errors are encountered

3) Spin up a container with image nginx:1.19.10 such that port 80 of the container can be connected from port 8080 of the host

4) Spin up a container with image redis:6.2.2 and mount volume /mnt/redis-data of host to the /data of the container

5) Create a dockerfile with base image centos:7 , and build an image with any sample application file

6) Create a bridge network called test-app and spin up nginx and redis containers in that network

**Kubernetes Questions:**

1) Create a pod called web-server with nginx:1.19.10 image

2) Expose port 80 of the web-server pod to be reachable within cluster

3) Create a single pod with below images:

i) nginx:1.19.10

ii) redis:6.2.2

4) Expose port 80 and 6379 of the above created pod such that the application can be connected from the outside world using node's IP address

5) Create a deployment web-deploy with nginx:1.19.10 image of 2 replica

6) Change the image of web-deploy to nginx:1.20.0 and record the change

7) Scale web-deploy to 5 replica

8) Create a persistent volume redis-pv with below specs:

i) hostpath /mnt/redis/data

ii) storage size 2Gi

iii) access mode - ReadWriteOnce

9) Create a persistent volume claim redis-pvc that claims redis-pv persistent volume

10) Create a pod redis which binds the redis-pvc to the path /data with image redis:6.2.2

11) Update the storage size of the redis persistent volume to 3Gi and record the change

12) Create an Ingress for web-deploy deployment with wildcard hostname