Practice question on Docker swarm

1. Create a cluster of 4 nodes -2 manager , 2 worker
2. Create a web based service named as web1
3. now scale the service with replica value of 4.
4. now check that services on manager node
5. Then check the status os service in worker node.

- 6. now Create one more service with name as web2 with option of replica through manager node.

docker service create --replicas 3 --name web2 your_image:tag

- 7. again check the status of the service and task in manager and worker nodes.

  Docker service ps web2 (in manager node)

  Docker ps (in worker node)

8. Now scale both services in one command with updated value.

   docker service update –replicas 5 web1 web2

9. Next is to rollback the web2 service and check its status.

   docker service rollback web2

Practice Q#2

To Create a service stack with the visualizer container.

What is Visualizer container?

Docker Stack

- It sits at a higher level than Docker containers and helps to manage the orchestration of multiple containers across several machines.
- Docker Stack is run across a Docker Swarm, which is essentially a group of machines running the Docker daemon, which are grouped together, essentially pooling resources.

- Stacks allow for multiple services, which are containers distributed across a swarm, to be deployed and grouped logically.
- The services run in a Stack can be configured to run several replicas, which are clones of the underlying container.
- This number will be maintained as long as the service is running.

- *Service* is the Swarm equivalent of a container. It represents one single container, running on one or more machines.
- *Stack* is the Swarm equivalent of Docker Compose. It represents a number of services, linked up to each other and deployed using a single configuration.

```
docker service create --name=app1\
    -p=8082:8080 \
    --constraint=node.role==manager \

    --mount=type=bind,src=/var/run/docker.sock,
    dst=/var/run/docker.sock \
    dockersamples/visualizer:stable
```

- Overall, this command creates a Docker service named “app1" using the "dockersamples/visualizer:stable" image, maps port 8080 of the container to port 8082 of the host system, constrains the service to run only on manager nodes, and mounts the Docker socket into the container to enable visualization of the Docker Swarm cluster.

Detail of this service

- This Docker command creates a Docker service named “app1" using the "dockersamples/visualizer:stable" image. Let's break down the command and its options:
- **docker service create**: This is the command to create a new Docker service.
- **--name=app1**: Specifies the name of the service as “app1".
- **-p=8082:8080**: Maps port 8080 of the container to port 8082 of the host system. This means that you can access the service running inside the container via port 8082 on the host.

- **--constraint=node.role==manager**: Constrains the service to only run on nodes that have the role of a manager in the Docker Swarm cluster. This ensures that the service only runs on manager nodes, providing more control over where the service is deployed.

- **--mount=type=bind,src=/var/run/docker.sock,dst=/var/run/docker.sock** : Mounts the Docker socket from the host system into the container at the same location. This allows the container to interact with the Docker daemon running on the host system, enabling it to visualize the Docker Swarm cluster.
- **dockersamples/visualizer:stable**: Specifies the Docker image to use for the service.
- In this case, it's the "dockersamples/visualizer" image tagged as "stable". This image provides a web-based visualization of the Docker Swarm cluster, allowing you to see the nodes and services running in the cluster.

Practice Q#3

- Explore the docker service update command in detail.

- Explore the example of each option and then prepare one word file which includes all options with examples.

Practice Q#4

- 1. Create a Yaml File for docker service
  (same for Practice Q#2)

- 2. Then deploy it on docker swarm cluster.

Solution:

```
version: '3.8'

services:
  app1:
    image: dockersamples/visualizer:stable
    ports:
      - "8081:8080"
    deploy:
      placement:
        constraints:
          - node.role == worker
    volumes:
      - type: bind
        source: /var/run/docker.sock
        target: /var/run/docker.sock
```

- Then run the following code on manager node in cluster

```
docker stack deploy --compose-file
dockercompose.yml app1
```

Working of the command

- The command docker stack deploy --compose-file dockercompose.yml app1 will deploy a Docker stack based on the configuration defined in the dockercompose.yml file.

Here's what each part of the command does:

- docker stack deploy: This command is used to deploy a stack defined in a Compose file to the Swarm cluster.

- --compose-file dockercompose.yml: Specifies the path to the Compose file (dockercompose.yml) that contains the stack configuration.
- app1: Specifies the name of the stack. In this case, the stack will be named app1.
- When you run this command, Docker Swarm will read the dockercompose.yml file, create the necessary services and networks as defined in the file, and deploy them onto the Swarm cluster under the specified stack name app1