**DockerSwarm:**

**Docker swarm is the product of docker. It is an orchestration tool which will help us to orchestrate the container deployment.**

**DockerSwarm is the tool which help us manage deploy the container across the multiple VM’s i.e., cluster of nodes.**

**Note: DockerSwarm is only used to create container using docker software other container software such as rocket etc. will not work with dockerswarm.**

**Dependencies: Master Node: Docker software installed up and running**

**Workernode: Docker software installed up and running**

**How to create cluster using Docker Swarm?**

**Master Node:  
To initiate the process run the below command on master node**

**Command: docker swarm init (It will initiate the dockerswarm process for creating cluster)**

**Note: once the process is completed you will get a command for joining the slave(workernode) to the cluster. Copy everything from docker swarm join along with the token it got generate and paste it in the slave vm for it to join the docker swarm cluster**

**Command: docker swarm join-token ( Run this on slave/workernode)**

**How to create a service in dockerswarm cluster?**

**Masternode:**

**To create a service on docker swarm run below command on the master node:**

**Command: docker service create –name svc1(Name of your choice) –replicas 4(No of your choice it will create replicas of the container across the cluster) -p 81:80(Binding port) nginx (Name of the image)**

**Once you run the above command it will create a service and that will create four containers running across the cluster both on master and the slave node.**

**Note: Service port 81 will be available across the cluster**

**Command to list out the services created on the master node:**

**Command: docker service ls**

**[centos@ip-172-31-46-136 ~]$ sudo docker service ls**

**ID NAME MODE REPLICAS IMAGE**

**rnii07v0bplc svc1 replicated 2/2 nginx:latest**

**[centos@ip-172-31-46-136 ~]$**

**Command to check the number of nodes attached to the cluster:**

**Command : docker node ls**

**[centos@ip-172-31-46-136 ~]$ sudo docker node ls**

**ID HOSTNAME STATUS AVAILABILITY MANAGER STATUS**

**gvsy4umnpxnrtdm1v7g541y7d \* ip-172-31-46-136.ap-south-1.compute.internal Ready Active Leader**

**sq4wh5h1gowp1ifiedoole6n6 ip-172-31-44-99.ap-south-1.compute.internal Ready Active**

**[centos@ip-172-31-46-136 ~]$**

**Command to delete the service created on master node:**

**Command: docker service rm serviced**

**[centos@ip-172-31-46-136 ~]$ sudo docker service rm rnii07v0bplc**

**rnii07v0bplc**

**[centos@ip-172-31-46-136 ~]$ date**

**Wed Dec 21 07:36:50 UTC 2022**

**[centos@ip-172-31-46-136 ~]$**

**Note: Deleting service will delete all the containers running across the cluster**

**Scaling up and scaling down containers using Docker Swarm:**

**Using docker swarm we can easily scale up or scale down our containers running across the cluster**

**Please run the below command to scale up the container across the cluster:**

**Command: docker service scale svc1(NameOfTheService)=10(No of your choice)**

**Please run the below command to scale down the container across the cluster:**

**Command: docker service scale svc1(NameOfTheService)=2(No of your choice)**

**Security with DockerSwarm:**

**As we have seen under services when we delete the service everything along with that service will be deleted. In order to prevent any unauthorize access we can use a feature of docker swarm to lock the access to the services created.**

**Command: docker swarm update –autolock=true( It will generate a password for us, Store the password some place safe because to unlock we require it)**

**To unlock the swarm use below command:**

**Command: docker swarm unlock (It will prompt you to enter the password enter the password which got generated earlier)**