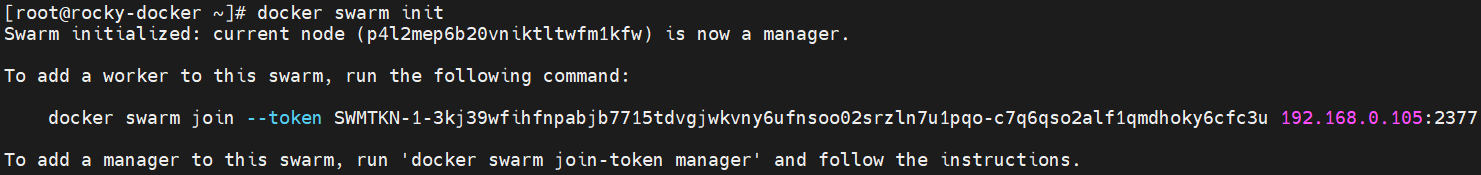
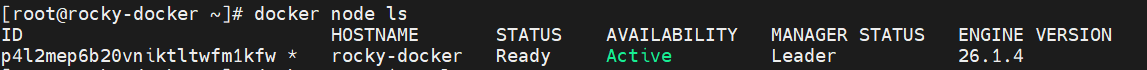
1. **Docker swarm provides us cluster of docker hosts.**
2. **Suppose one docker host gets down on which your containers are running then docker swarm will check for other available docker host and start those containers on it.**
3. **In this concept there are managers and workers. Managers can also work as worker sometimes. i.e you can do the configuration in such a way that manager can work like both worker and manager.**
4. **Managers are always sync with each other.**
5. **Worker performs the tasks given by master(manager). On worker, there is an agent installed which sends current state of tasks to master.**
6. **Transport layer security (TLS) use between masters and workers communication. Master is having the certificate and worker has token.**
7. **#docker swarm init**

**This is a very first command to run on docker master server to make docker master.**

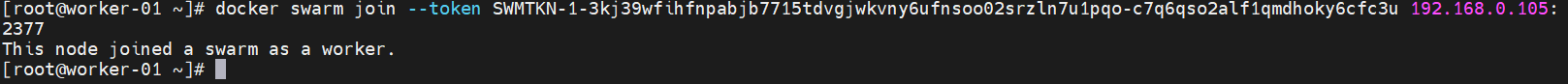
****

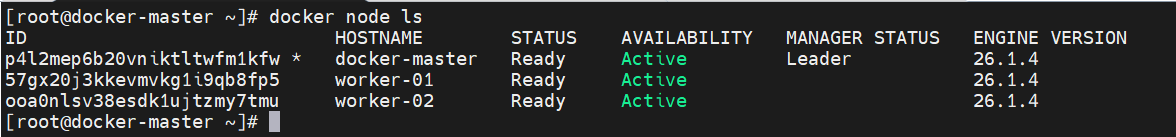
1. **#docker node ls**

**Command to tell you how many nodes are available in your cluster.**

****

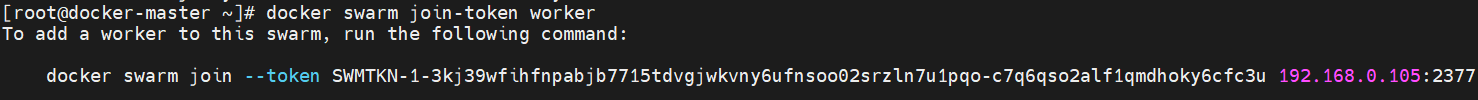
1. **We have to run docker swarm join command as below on worker node so that worker will join the cluster.**

****

****

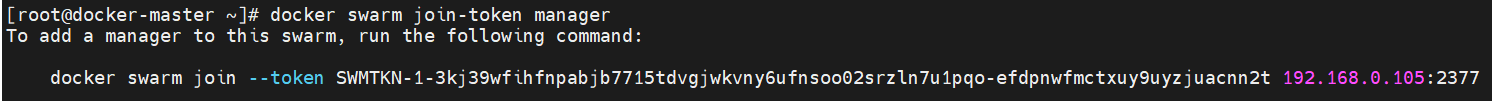
1. **#docker swarm join-token worker**

**To get worker token**

****

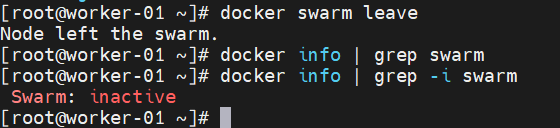
1. **#** **docker swarm join-token manager**

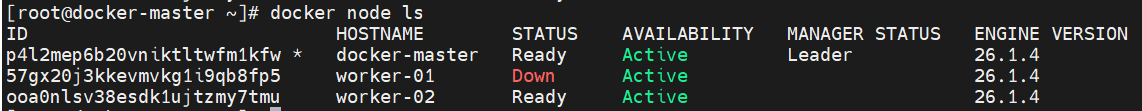
**To get master token. Suppose your setup is expanding and want add more masters in this setup.**

****

1. **#docker swarm leave**

**Need to run this command on worker server to get it out from cluster.**

****

****

1. **#docker node rm worker-01**

**After get out from cluster you need to completely remove it. So run above command on master server.**

1. **#docker node inspect worker-01 | less**

**To get details about the worker server or node**

****

1. **#docker node promote worker-01 worker-02**

**To act your worker nodes as manager or master.**

****

**Note: As per Docker recommendations, number of master servers should be in odd number i.e 1,3,5 etc**

1. **#docker node demote worker-01 worker-02**

**To demote and act as a worker like before.**

1. **#docker service create -d ubuntu ping 192.168.0.107**

**To build container on worker node and run any command. Here took ping command as a example.**

****

**Explore more commands**

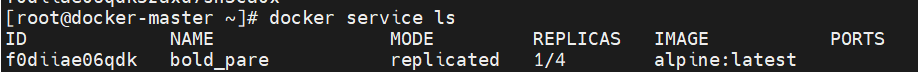
**#docker service --help**

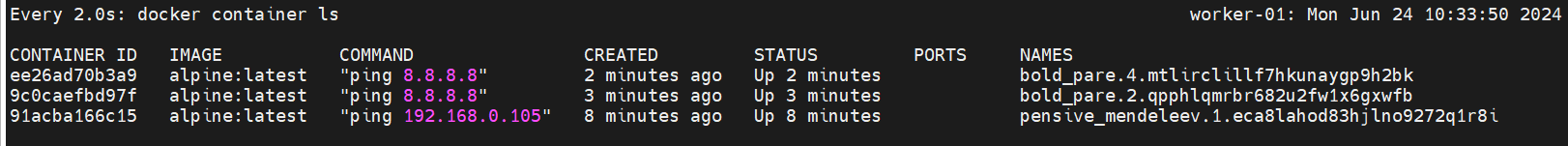
**You can check this task assigned to which worker node by running below command on worker node.**

**#docker container ls**

1. **#docker service create -d --replicas 4 alpine ping 8.8.8.8**

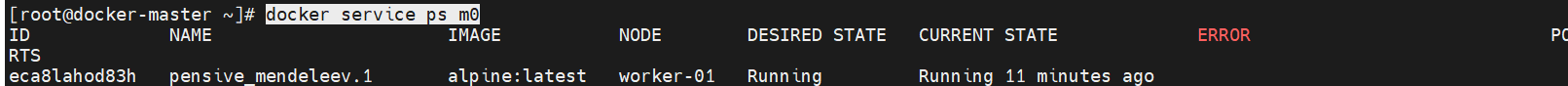
**If you wish to create multiple replicas.**

****

****

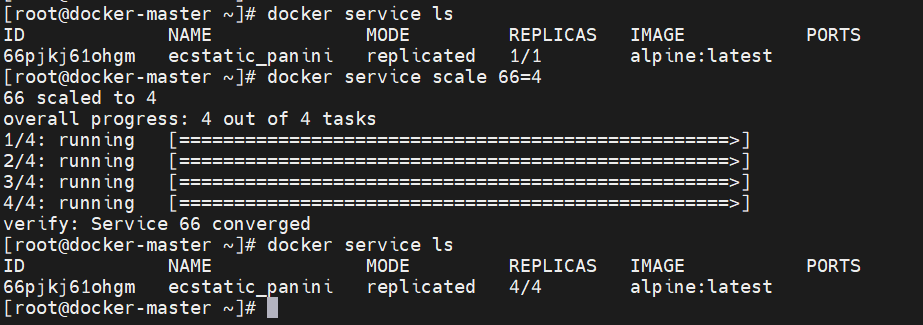
1. **#** **docker service ps m0(service id)**

**It will show the details about the container or services running on respective worker. Run this command on master server.**

****

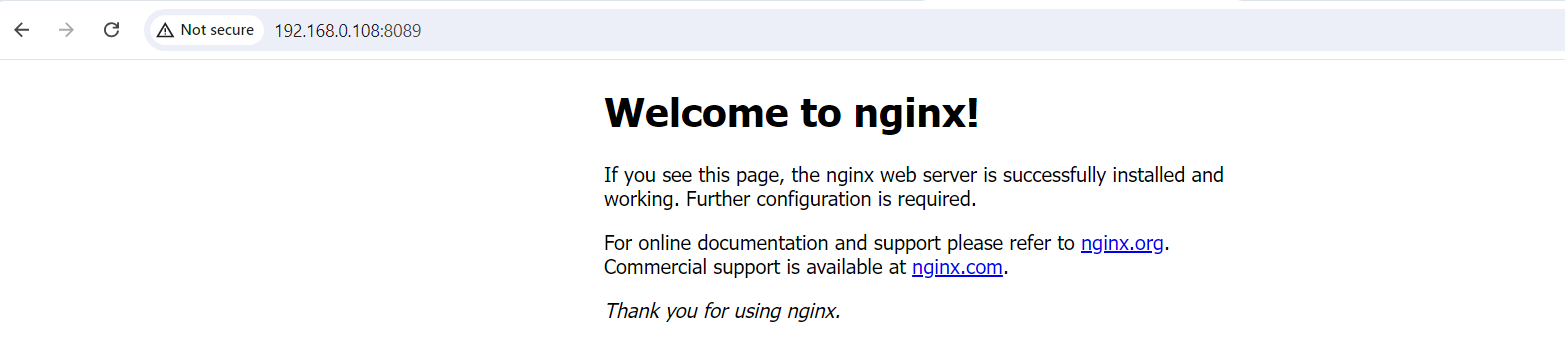
1. **If you delete any container from worker server then master server will immediately create another container as per replica mentioned.**
2. **#docker service scale 66=4**

**To scale number of containers**

****

1. **#docker service create -d -p 8089:80 nginx**

**To assign port. Please note that you can now access nginx by using any workers or master ip. Because it is cluster.**

****

1. **#docker service create --mode=global nginx ping 8.8.8.8**

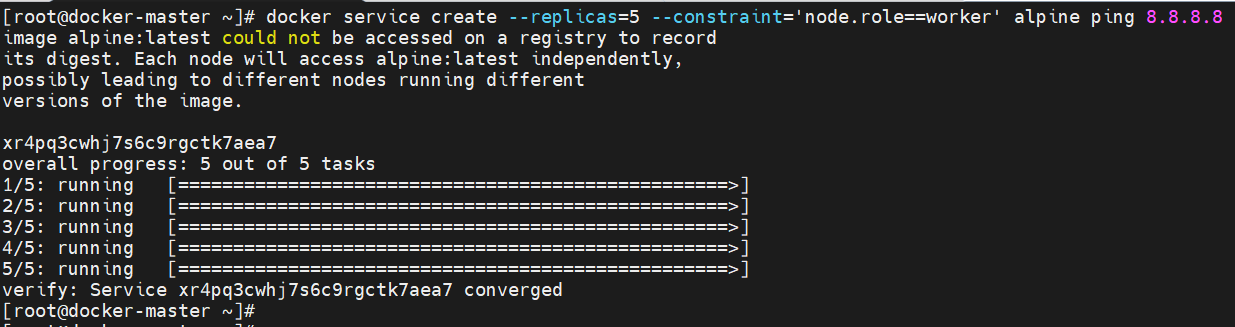
**When you create service in global mode. Then container will get created on your future worker node as well(which you will add in cluster future)**

1. **#docker service create --replicas=3 --constraint='node.role==manager' alpine ping 8.8.8.8**

**To create container on manager node only.**

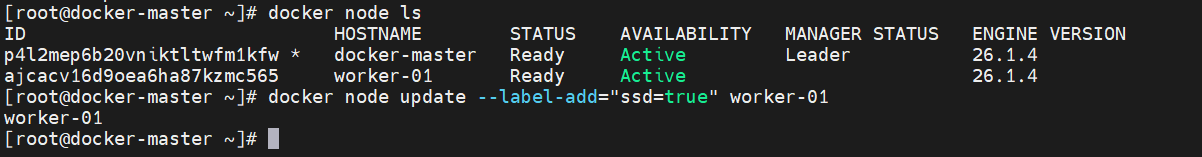
**#docker service create --replicas=5 --constraint='node.role==worker' alpine ping 8.8.8.8**

**To create container on worker nodes only.**

****

1. **# docker node update --label-add="ssd=true" worker-01**

**To do labeling on worker node.**

****

**If you create service by using label then container will get created on that particular worker node only. Here we have taken example of worker node where SSD has been used hence labeling done as SSD on it.**

**# docker service create --constraint="node.labels.ssd==true" --replicas=3 -d alpine ping 8.8.8.8**

**Please note that docker will not balance your containers if you do same labeling on other worker node post service creation. Please do necessary labeling before creating service.**

1. **You can assign label in two ways. Node level and engine level. Do google for engine level labeling.**
2. **There are 3 types of availability of nodes.**

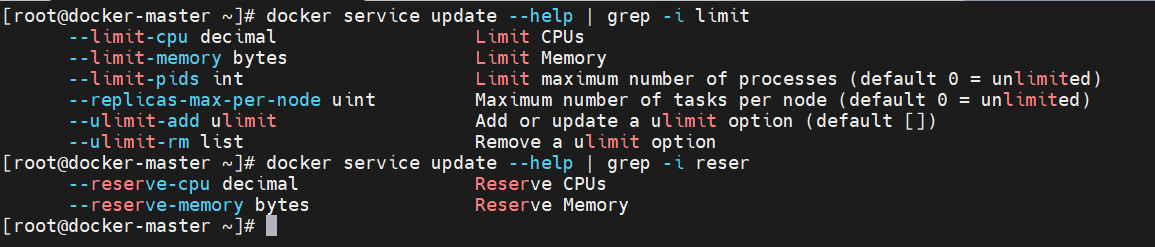
**Active: Node is ready to take new task from manager**

**Pause: Node is not ready to take new task**

**Drain: Drain node containers will be shifted in other working nodes.**

****

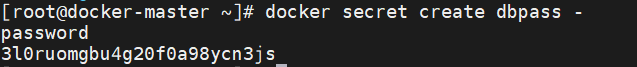
1. **Check below options on google. This is also imp.**

****

1. **# docker secret create dbpass –**

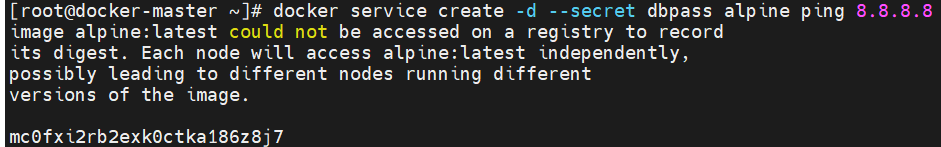
**Enter pass & ctrl + D**

**To create secrets. It is useful in storing db password, ssh keys, aws keys etc.**

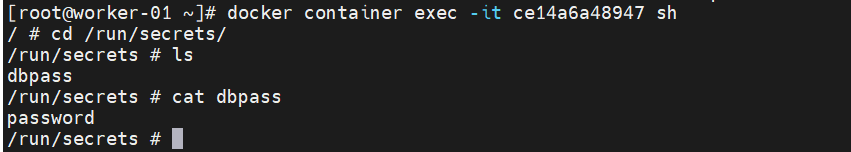
****

**Now how to use this secret.**

**#docker service create -d --secret dbpass alpine ping 8.8.8.8**

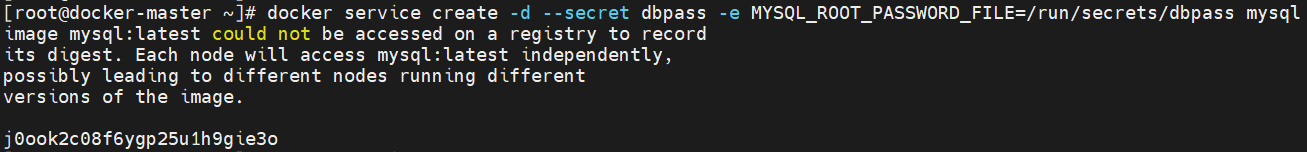
****

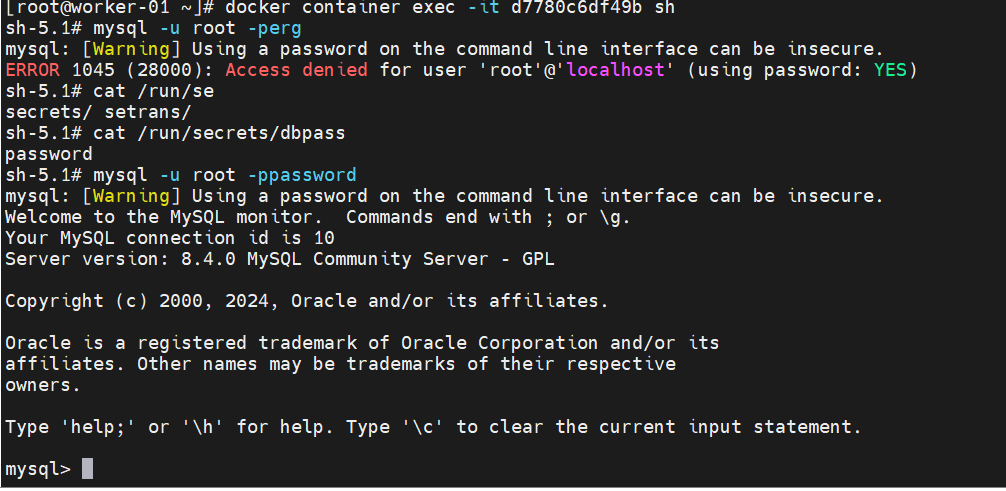
**This secret stores in /run/secret on worker server.**

****

**Sample example of using secrets in database container.**

**# docker service create -d --secret dbpass -e MYSQL\_ROOT\_PASSWORD\_FILE=/run/secrets/dbpass mysql**

****

****