Docker commands

* Execute the following command to remove the old versioned docker software:

$ sudo apt-get remove docker docker-engine docker.io containerd runc

Verfiy docker installation

$ sudo docker run hello-world

### Step 4: Create a Docker Image

* Once you have created and edited the main.py file and the Dockerfile, create your image contain your application by running the following command:

$ sudo docker build -t python-test .

-t : name of your image

### Step 5: Run Docker Container

* Once the image is created, your code is ready to launch.

$ sudo docker run python-test

* docker –version
* vi dockerfile

$ docker image ls

Docker run commands

$ docker run <image\_name>  
To give name of container  
$ docker run --name <container\_name> <image\_name>

### ****Docker Pull****

$ docker pull <image\_name>

### ****Docker PS****

This command (by default) shows us a list of all the running containers. We can use various flags with it.

* **-a flag:** shows us all the containers, stopped or running.
* **-l flag:**shows us the latest container.
* **-q flag**: shows only the Id of the containers.

$ docker ps [options..]

### ****Docker Stop****

This command allows you to stop a container if it has crashed or you want to switch to another one.

$ docker stop <container\_ID>

### ****Docker Start****

Suppose you want to start the stopped container again, you can do it with the help of this command.

$ docker start <container\_ID>

### ****Docker rm****

$ docker rm {options} <container\_name or ID>

docker remove an image

### Docker RMI

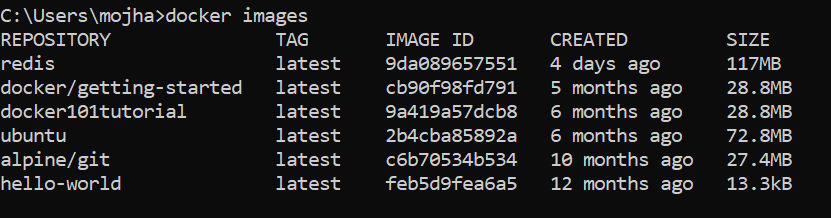
To delete the image in docker. You can delete the images which are useless from the docker local storage so you can free up the space

docker rmi <image ID/ image name>

### ****Docker Images****

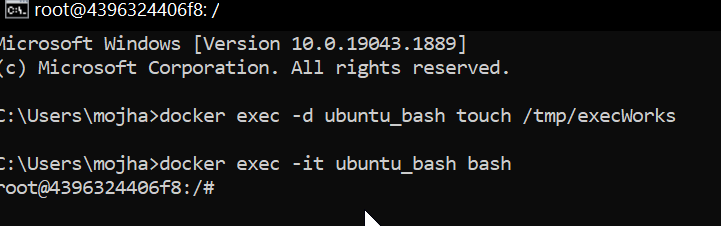
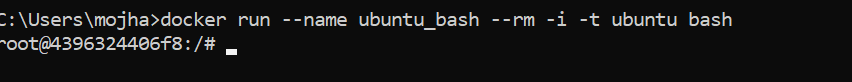
Lists all the pulled images which are present in our system.

$ docker images



### ****Docker exec****

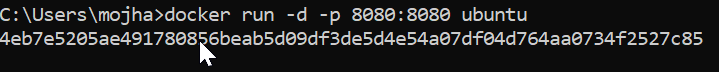
$ docker exec {options}



### ****Docker Ports (Port Mapping)****

In order to access the [docker container](https://www.geeksforgeeks.org/containerization-using-docker/) from the outside world, we have to map the port on our host( Our laptop for example), to the port on the container. This is where port mapping comes into play.

$ docker run -d -p <port\_on\_host>   
<port\_on\_container> Container\_name



Docker push

docker push <Image name/Image ID>

docker build

docker build -t image\_name:tag

### Docker Stop

docker stop container\_name\_or\_id

### Stop Multiple Containers

docker stop container1 container2 container3

### Docker Restart

docker restart container\_name\_or\_id

### Docker Inspection

docker inspect container\_name\_or\_id

### Docker Commit command

docker commit container\_name\_or\_id new\_image\_name:tag

## **Docker Basic Command**

Following are the some of the docker basic commands

1. **docker images:** Docker images will list all the images which are pulled or build in that docker host.
2. **docker pull:**Docker pull will the docker images from the dockerhub.
3. **docker run:**Docker run will run the docker image as an container.
4. **docker ps:**Docker run will list all the containers which are running in the docker host.
5. **docker stop:**Docker stop will stop the docker container which are already running.
6. **docker rm:**Docker rm command will remove the containers which are in the stop condition.

## Docker Commands List

Following are the docker commands which listed form build and Docker image to running it an Docker container and attaching the docker volumes to it.

### Docker Image Command

1. **docker build command:** It will build Docker images by using the **Dockerfile.**
2. **docker pull command:** Docker pull command will pull the **Docker image** whcih is avalible in the **dockerhub.**
3. **docker images command:**It will list all the images which are pulled and build in the docker host.
4. **docker inspect command:**It will helps to debug the docker image if any errors occurred while building an image or pulling the image.
5. **docker push command:**Docker command will push the docker image into the Dockerhub.
6. **docker save command:**It will save the docker image in the form of dockerfile.
7. **docker rmi command:**It will remove the docker image.

### Docker Container Command

1. **docker attach command:** Connecting to an Existing Container.
2. **docker ps command:**To list the running containers.
3. **docker container inspect infinite Command:**To Inspect the Docker containers.
4. **docker exec command:**To execute the commands in the running containers.
5. **docker cp command:** To copy the file from docker host to the docker containers,

### 2. Docker Command Not Found

*Docker command not found error will get if the docker cli was not installed in your system.*

### 3. Docker Command To list the All Containers

*To list all the containers in the docker you can use the****docker pa -a.***

### 4. Docker Command To Build An Image

*To build the docker image you can use the following command.*

***docker build -t <image name>:<tag> <path to Dockerfile>***

### 5. Docker Command To Remove An Image

*Docker Command to remove the docker images was*

***docker rmi <image name>:<tag>***

### 6. Docker Command Cheat Sheet

*To read the docker cheat sheet refer to*[*Docker Cheat Sheet – Most Important Docker Commands.*](https://www.geeksforgeeks.org/docker-cheat-sheet/)

### 7. Docker Command to Command to check the Running Containers

*docker ps is the command to check the running containers.*

### 8. What is the Docker file with all commands?

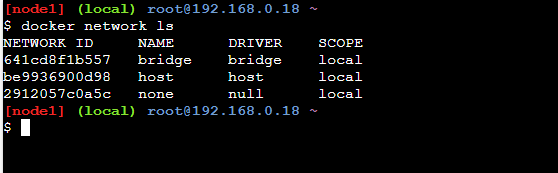
*Dockerfile is set of instruction to build the Docker image*

We can use the following command to create docker volume -:

docker volume create <volume\_name>

To remove a volume, we can use the following command -:

docker volume prune

* docker network ls // see docker network list
* sudo systemctl status docker // status of docker
* sudo systemctl enable docker --now

// enabling docker services

## **Inspecting The Bridge Network**

* After you have created the [Docker Container,](https://www.geeksforgeeks.org/containerization-using-docker/) check whether it is running or not.

sudo docker container ls

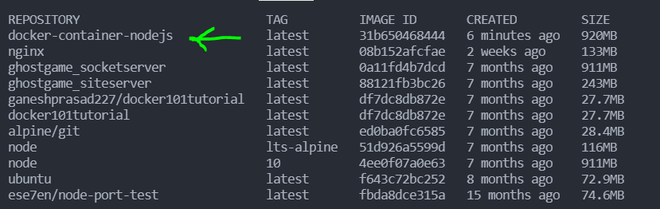
Inspecting the Bridge Network

* Now, initialize the node project using the following command.

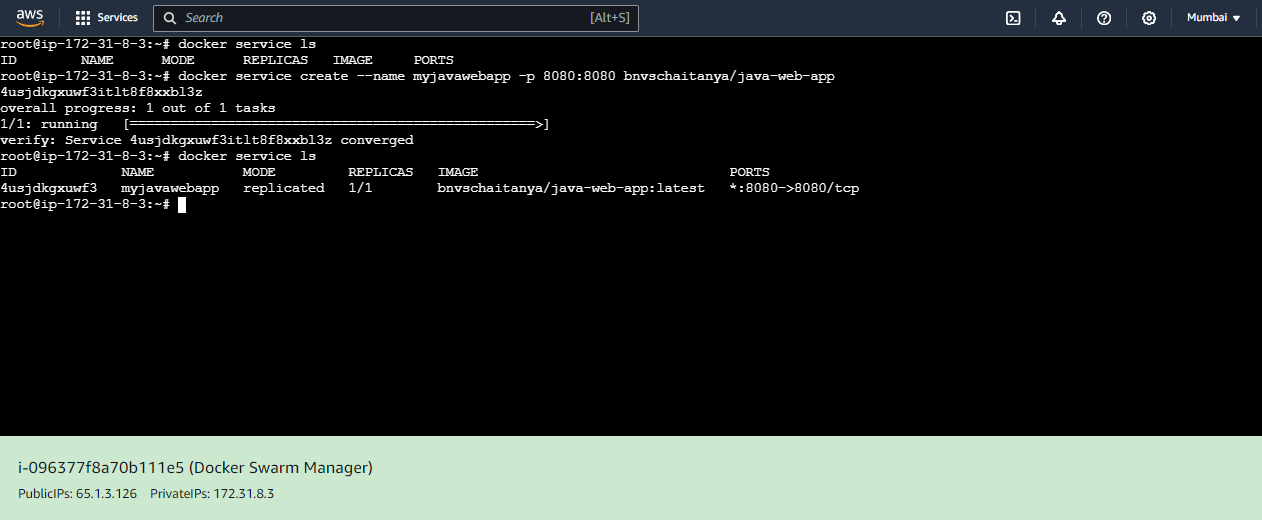
npm init

* Confirm that the image has been created.

docker images



>docker service ls



## **Docker Swarm Mode CLI Commands**

The following are the docker swarm mode CLI commands:

* **docker swarm init:**This command is used to initialize the swarm.

docker swarm init [OPTIONS]

example: docker swarm init –advertise-addr 192.168.99.121

* **docker swarm join:**By using this command you can join a node to a swarm. The node joins as a manager node or worker node based on the token you pass with the –token flag.

docker swarm join [OPTIONS] HOST:PORT

docker swarm join –token swmtkn -1 3pu6hszjas19xyp7ghgosyx9

docker node ls

//this node joined a swarm as a manager

* **docker service creates:** This is a cluster management command, and must be executed on a Swarm manager node.

docker service create [OPTIONS] IMAGE [COMMAND] [ARG...]

docker service create --name redis --secret secret.json redis:3.0.6

4cdgfyky7ozwh3htjfw0d12qv

* **docker service inspects:**This command is used to inspect the particular service and all the details will display in JSON format.

docker service inspect [OPTIONS] SERVICE [SERVICE...]

* **docker service ls:** This command is used to see the complete list of all the services in that network.

docker service ls [OPTIONS]

* **docker service rm:** This command is used to remove the specific service you want to remove.

docker service rm SERVICE [SERVICE...]

**Create a volume:**

$ docker volume create my-vol

**List volumes:**

$ docker volume ls

$ docker volume inspect my-vol

**Remove a volume:**

$ docker volume rm my-vol

**Backup volume steps**

### Step 1: Create a Docker Volume

* A Docker volume can be created using the docker volume create command.

docker volume create my\_volume

### Step 2: Use the Volume in a Docker Container

* You can run a Docker container and mount the volume to a specific path within the container using the -v option.

docker run -d --name my\_container -v my\_volume:/data busybox

### Step 3: Add Data to the Volume (Optional)

* You might want to add some data to the volume to simulate a real-world scenario.

docker exec -it my\_container sh -c "echo 'Hello, GeeksforGeeks!' > /data/hello.txt"

### Step 4: Backup the Docker Volume

* To back up the Docker volume, you can use a temporary container to copy the volume data to a tar archive.

docker run --rm -v my\_volume:/data -v $(pwd):/backup busybox tar cvf /backup/my\_volume\_backup.tar /data

**In this command**

* **–rm:** Automatically remove the container when it exits.
* **-v my\_volume:/data:**This mounts the volume to /data inside the container.
* **-v $(pwd):/backup:** Mounts the current directory to /backup in the container where the backup file is located.
* **tar cvf /backup/my\_volume\_backup.tar /data:** This creates a tarball archive of the directory /data with the volume data within it and places it in /backup.

### Step 5: Verify the Backup

* Check the backup file to ensure it was created successfully:

ls -lh /home/ec2-user/backup/my\_volume\_backup.tar

### Step 6: Restore the Docker Volume (If Needed)

* To restore the Docker volume from the backup, create a temporary container and use the following command:

docker run --rm -v my\_volume:/volume -v $(pwd):/backup busybox tar xzf /backup/backup.tar.gz -C /volume