

Docker Commands

- List All Running Docker Containers
`docker ps`
- List All Docker Containers
`docker ps -a`
- Start a Docker Container
`docker start <container name>`
- Stop a Docker Container
`docker stop <container name>`
or
`docker kill <container name>`
- Kill All Running Containers
`docker kill $(docker ps -q)`
- View the logs of a Running Docker Container
`docker logs <container name>`
- Delete All Stopped Docker Containers
Use -f option to nuke the running containers too.
`docker rm $(docker ps -a -q)`
- Remove a Stopped Docker Container
`docker rm <container name>`
- List all Docker Images
`docker images -a`
or
`docker images`
- Run A Docker Image
`docker run <image name>`
- Build a docker image
From the directory of the Dockerfile run:
`docker build -t <tag name>`
- Remove a Docker Image
`docker rmi <image name>`
- Delete All Docker Images
`docker rmi $(docker images -q)`
- Delete All Untagged (dangling) Docker Images
`docker rmi $(docker images -q -f dangling=true)`

- Remove Dangling Volumes
`docker volume rm -f $(docker volume ls -f dangling=true -q)`
- Parameter that tells docker to run the container as a background process
`-d`
 Example:
`docker run -d <image name>`
- Map a Host Port to a Container Port
`-p <host port>:<container port>`
 Example:
`docker run -p 8080:8080 <image name>`
- SSH Into a Running Docker Container
 Okay not technically SSH, but this will give you a bash shell in the container.
`sudo docker exec -it <container name> bash`
- Share Storage on a Host System with a Docker container
`-v <host path>:<container path>`
 Example:
`docker run -v <host path>:<container path> <image name>`

Docker Compose Commands

- Use Docker Compose to Build Containers
 Run from directory of your docker-compose.yml file.
`docker-compose build`
- Use Docker Compose to Start a Group of Containers
 Use this command from directory of your docker-compose.yml file.

`docker-compose up -d`

This will tell Docker to fetch the latest version of the container from the repo, and not use the local cache.

`docker-compose up -d --force-recreate`

This can be problematic if you're doing CI builds with Jenkins and pushing Docker images to another host, or using for CI testing. I was deploying a Spring Boot Web Application from Jenkins, and found the docker container was not getting refreshed with the latest Spring Boot artifact.

```
#stop docker containers, and rebuild
docker-compose stop -t 1
docker-compose rm -f
docker-compose pull
docker-compose build
docker-compose up -d
```

- Follow the Logs of Running Docker Containers With Docker Compose
docker-compose logs -f
- Save a Running Docker Container as an Image
docker commit <image name> <name for image>
- Follow the logs of one container running under Docker Compose
docker-compose logs pump <name>

Docker Swarm Commands

- Is Docker Swarm automatically enabled?
No, by default, Docker Swarm is not available
- Types of Nodes in a Docker Swarm
Manager and worker
- Enable the First Node of a Docker Swarm
docker swarm init
- List Running Services
docker service ls
- Add a Node to a Swarm Cluster
docker swarm join --token <token> --listen-addr <ip:port>
- Can manager nodes run containers?
Yes, manager nodes normally run containers
- Retrieve the Join Token
docker swarm join-token
- List Nodes in a Cluster
docker node ls
- Can you run a 'docker node ls' from a worker node?
No. Docker Swarm commands can only be from manager nodes
- List Services in a Docker Swarm
docker service ls
- List Containers in a Service
docker service ps <service name>
- Remove a Service
docker service rm <service name>
- Remove a Node from a Swarm Cluster
docker node rm <node name>

- Promote a Node from Worker to Manager
docker node promote <node name>
- Change a Node from a Manager to a Worker
docker node demote <node name>
- Map a Host Port to a Container Port
- p <host port>:<container port>
Example:
docker run -p 8080:8080 <image name>