Basic:

docker run <iamge name>

docker run --name <any name> <image name>(assign a name to the container under this image)

docker pull <image name>(only pull the image but not running it)

docker image ls (list all the running images)

docker rm<container name>(remove stopped or exited container)

docker rmi <image name>(remove the image but must make sure all dependent containers are removed)

docker create XXX (this will create a container with ID)

docker start containerID(this will start the container image)

docker start -a containerID(this will start the container image and get all the result print to console)

docker system prune(this will remove all the stopped containers)

docker logs <containerID>(get logs of the container, not restarting)

docker stop <containerID | container name>(takes 10 secs to kill the container if the container does not know the signal)

docker kill <containerID>(instantly dead)

docker ps(get containers)

docker ps --all (get all containers)

docker exec -it <ContainerID> <Command> (execute an additional command in a container, -it allow us to provide inpu to container)

docker exec -it <ContainerID> bash/powershell/zsh/sh (joins already running container's bash prompt.)

docker run -it <image name> bash/powershell/zsh/sh (starts a new container with prompt.)

docker run -it<image name>(attach to terminal and interactive mode)(with prompt and output)

docker run -d <image name>(to run this container in background, detached)

docker attach <containerID>(canbe first few letters, attach back to specific container)

ctrl+C or ctrl+D or type exit to exit to master

Docker Run:

Docker run <image:version>(run older version of the image)

Run-PORT mapping:

Docker run -p <80:5000 > <image name>(so that user can access to the port 80 of this docker host’s IP address, port 80 route to port 5000 inside the container)

Run-Volume mapping

Docker run -v <dockerhostDir : insideContainerDir> <image name>

Ex: docker run -v /opt/datadir : /var/lib/mysql mysql

↑ ↑

Dir in docker host Dir inside container

So all the data will be stored in outer volume, thus if container get deleted , data still there.

Inspect Container:

Docker inspect <container name>

Container logs:

Docker logs <container name>