**Docker Commands:**

docker build : build an image from the Docker file

docker image ls : lists all the images that are locally stores with the docker engine

docker image rm : Delete an image from the local image store

docker pull : Pull an image from a registry

docker push : Push an image to a registry

docker commit : create a new image from a container’s changes

docker create : create a new container

docker kill : kill one or more running containers

docker logs : fetch the logs of a container

docker run : runs a command in a new container

docker service : manage services

docker stop : stop one or more running containers

docker container : manage containers

docker network : manage networks

**What is a Docker?**

Docker is an open platform for developing, shipping, and running applications. Docker enables us to separate our applications from our infrastructure so we can deliver software quickly. With Docker, we can manage our infrastructure in the same ways we manage our applications. By taking advantage of Docker’s methodologies for shipping, testing, and deploying code quickly, we can significantly reduce the delay between writing code and running it in production.

**Why Docker?**

**Fast, consistent delivery of our applications**: Docker streamlines the development lifecycle by allowing developers to work in standardized environments using local containers which provide our applications and services. Containers are great for continuous integration and continuous delivery (CI/CD) workflows.

**Responsive deployment and scaling:** Docker’s container-based platform allows for highly portable workloads. Docker containers can run on a developer’s local laptop, on physical or virtual machines in a data center, on cloud providers, or in a mixture of environments.

**Running more workloads on the same hardware**: Docker is lightweight and fast. It provides a viable, cost-effective alternative to hypervisor-based virtual machines, so we can use more of our compute capacity to achieve our business goals. Docker is perfect for high density environments and for small and medium deployments where you need to do more with fewer resources.

**Docker Architecture:**

Docker uses a client-server architecture. The Docker client talks to the Docker daemon, which does the heavy lifting of building, running, and distributing our Docker containers. The Docker client and daemon can run on the same system, or we can connect a Docker client to a remote Docker daemon. The Docker client and daemon communicate using a REST API, over UNIX sockets or a network interface.

