

Docker

Docker is a set of platforms as a service (PaaS) products that use the Operating system level virtualization to deliver software in packages called containers.

Docker Image

- It is a file, comprised of multiple layers, used to execute code in a Docker container.
- They are a set of instructions used to create Docker containers.

Docker Container

- It is a runtime instance of an image.
- Allows developers to package applications with all parts needed such as libraries and other dependencies.

Docker file

- It is a text document that contains necessary commands which on execution helps assemble a Docker Image.
- Docker image is created using a Docker file.

Registries and Repositories

- A registry is where we store our images.
- You can host your own registry, or you can use Docker's public registry which is called DockerHub.
- Inside a registry, images are stored in repositories. Docker repository is a collection of different docker images with the same name, that have different tags, each tag usually represents a different version of the image.

Docker Docker file

A Docker file is a text document that contains commands that are used to assemble an image. We can use any command that call on the command line. Docker builds images automatically by reading the instructions from the Dockerfile.

Docker file Instructions

The instructions are not case-sensitive but you must follow conventions which recommend to use uppercase.

Docker runs instructions of Dockerfile in top to bottom order. The first instruction must be **FROM** in order to specify the Base Image.

A statement begin with # treated as a comment. You can use RUN, CMD, FROM, EXPOSE, ENV etc instructions in your Dockerfile.

Chain RUN Instructions

- Each RUN command will execute the command on the top writable layer of the container, then commit the container as a new image.
- The new image is used for the next step in the Dockerfile. So each RUN instruction will create a new image layer.
- It is recommended to chain the RUN instructions in the Dockerfile to reduce the number of image layers it creates.

CMD Instructions

- CMD instruction specifies what command you want to run when the container starts up.

- If we don't specify CMD instruction in the Dockerfile, Docker will use the default command defined in the base image.
- The CMD instruction doesn't run when building the image, it only runs when the container starts up.
- You can specify the command in either exec form which is preferred or in shell form.

Docker Compose

- Docker compose is a very handy tool to quickly get docker environment up and running.
- Docker compose uses yaml files to store the configuration of all the containers, which removes the burden to maintain our scripts for docker orchestration.

Docker Network Types

- Closed Network / None Network
- Bridge Network
- Host Network
- Overlay Network

None Network

- Provides the maximum level of network protection.
- Not a good choice if network or Internet connection is required.

- Suites well where the container require the maximum level of network security and network access is not necessary.

Host Network

- Minimum network security level.
- No isolation on this type of open containers, thus leave the container widely unprotected.
- Containers running in the host network stack should see a higher level of performance than those traversing the docker bridge and ip tables port mappings.

Continuous Integration

- Continuous integration is a software engineering practice in which isolated changes are immediately tested and reported when they are added to a larger code base.
- The goal of Continuous integration is to provide rapid feedback so that if a defect is introduced into the code base, it can be identified and corrected as soon as possible.

How Swarm cluster works

- To deploy your application to a swarm, you submit your service to a manager node.
- The manager node dispatches units of work called tasks to worker nodes.
- Manager nodes also perform the orchestration and cluster management functions required to maintain the desired state of the swarm.

- Worker nodes receive and execute tasks dispatched from manager nodes.
- An agent runs on each worker node and reports on the tasks assigned to it. The worker node notifies the manager node of the current state of its assigned tasks so that the manager can maintain the desired state of each worker.