Docker

By,
Harshil Kanakia

Outline

- Basics of Docker
- Docker installation
- Understanding Docker image and container
- Commands related to Docker image and container

Docker

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

The Docker platform

- Docker provides the ability to package and run an application in a loosely isolated environment called a container.
- Docker provides tooling and a platform to manage the lifecycle of your containers:

1. Develop your application and its supporting components using containers.
2. The container becomes the unit for distributing and testing your application.
3. When you’re ready, deploy your application into your production environment, as a container or an orchestrated service. This works the same whether your production environment is a local data center, a cloud provider, or a hybrid of the two.

Use of Docker

- Fast, consistent delivery of your applications

  Consider the following example scenario:

1. Your developers write code locally and share their work with their colleagues using Docker containers.
2. They use Docker to push their applications into a test environment and execute automated and manual tests.
3. When developers find bugs, they can fix them in the development environment and redeploy them to the test environment for testing and validation.
4. When testing is complete, getting the fix to the customer is as simple as pushing the updated image to the production environment.

Use of Docker (Contd..)

- Responsive deployment and scaling

1. 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.

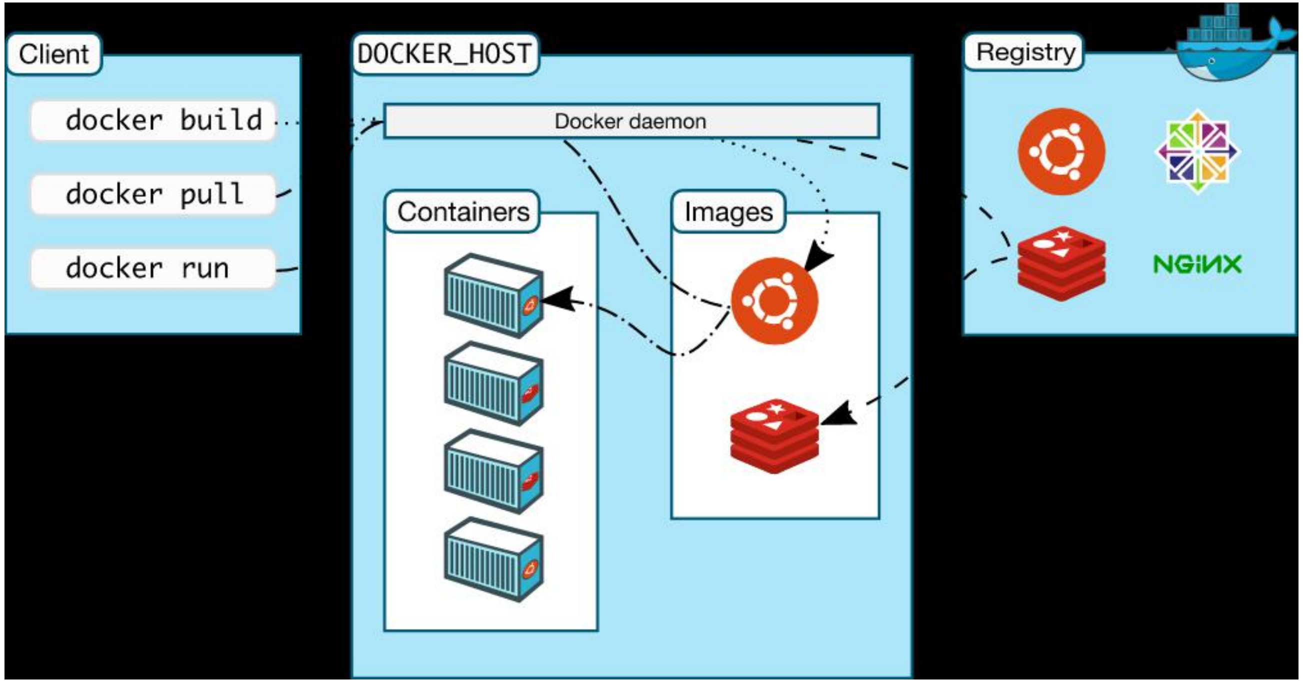
2. Docker’s portability and lightweight nature also make it easy to dynamically manage workloads, scaling up or tearing down applications and services as business needs dictate, in near real time.

Use of Docker (Contd..)

- 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 you can use more of your compute capacity to achieve your business goals

Docker Architecture



P.C. : docs.docker.com

Docker Components

- Docker daemon
- Docker client
- Docker registries
- Docker objects
    1. Images
    2. Containers

Docker Daemon

- The Docker daemon (dockerd) listens for Docker API requests and manages Docker objects such as images, containers, networks, and volumes.
- A daemon can also communicate with other daemons to manage Docker services.

Docker client

- The Docker client (docker) is the primary way that many Docker users interact with Docker.
- When you use commands such as docker run, the client sends these commands to dockerd, which carries them out.
- The Docker client can communicate with more than one daemon.

Docker registries

- A Docker registry stores Docker images.
- Docker Hub is a public registry that anyone can use, and Docker is configured to look for images on Docker Hub by default. You can even run your own private registry.
- When you use the docker pull or docker run commands, the required images are pulled from your configured registry. When you use the docker push command, your image is pushed to your configured registry.

Docker Object : Images

- An image is a read-only template with instructions for creating a Docker container.
- Often, an image is based on another image, with some additional customization. For example, you may build an image which is based on the ubuntu image, but installs the Apache web server and your application, as well as the configuration details needed to make your application run.

Docker Object : Containers

- A container is a runnable instance of an image.
- You can create, start, stop, move, or delete a container using the Docker API or CLI.
- You can connect a container to one or more networks, attach storage to it, or even create a new image based on its current state.

What will be there in this experiment?

- Docker Installation
- Downloading Dockers images.
- Uploading the images in Docker Registry
- Running commands in container.
- Running multiple containers.

Docker Installation

- Step 1: Update Repositories

Before beginning, it’s a good idea to update the local database of your software to make sure that there is access to the latest revisions.

To do so, run the following command on the terminal

```
sudo apt-get update
```

Docker Installation (Contd..)

- Step 2: Remove prior installations

  Next, you need to make sure that your system does not have any prior Docker software installation that may be outdated.

For that, run:

```
sudo apt-get remove docker docker-engine docker.io
```

Docker Installation (Contd..)

- Step 3: Install Docker

To install Docker on Ubuntu, run the following command in the terminal:

```
sudo apt-get install docker.io
```

Docker Installation (Contd..)

- Step 4: Start and automate Docker

  In order for Docker to be up and running at system startup, run the following commands one by one:

```
sudo systemctl start docker

sudo systemctl enable docker
```

Docker Installation (Contd..)

- Step 5: Check version

    To verify whether the installation has been successful, it is a good idea to verify the Docker version number installed. For that, run:

```
docker --version
```

Downloading Dockers images

- https://hub.docker.com/ is a repository where you will find docker images.
- Following is a command to download MongoDB image

```
sudo docker pull mongo
```

- Following is a command to check for what all images are downloaded

```
sudo docker images
```

- Following is a command to remove images

```
sudo docker rmi image-id/name
```

Uploading the images in Docker Registry

- Following is the command to upload the image in Docker registry

```
docker push docker/getting-started
```

Running commands in container

- Following is the command to check what all containers are running:

```
sudo docker ps
```

- Following is the command to check what all containers are running and not running/exited

```
sudo docker ps -a
```

Running commands in container

- Following is a command to run docker container,

```
sudo docker run ubuntu
```

- Following is a command to remove container,

```
sudo docker rm container-id/name
```

Case Study

- Download the app : https://docs.docker.com/get-started/02_our_app/
- Update the app : https://docs.docker.com/get-started/03_updating_app/
- Share the app : https://docs.docker.com/get-started/04_sharing_app/
  Download the shared app and run it again.
- Multi Container App : https://docs.docker.com/get-started/07_multi_container/

Reference: https://docs.docker.com/get-started/

:) :) Thank You! :) :)