Before we get started I would like to introduce a few terms that we will using in this article on Docker:

- Image: Read-only template with intructions to create a Docker container. Information stored in image is in JSON format.
- Container: A runnable instance of an image. It is configurable and isolated from other containers.
- Service: allows you to define the expect state of an application.
- Dockerfile: A textfile with instructions to create an image.
- Docker Store: A public repository of Docker images

Docker Tutorial Series Part 1: Images, Layers and Dockerfile

This article is first in a series of articles on Docker. Today, we are going to introduce the core concepts of Docker. Then we will build our first Docker image using Dockerfile and then instantiate a container.

If you want to get a brief overview of containers and virtual machines then visit this link:

Age of Docker: Containers vs Virtual Machines

Virtual Machines and Containers perform the same task—they isolate an application and its dependencies, allowing it...medium.com

Docker is a platform that allows you to separate your application from your infrastructure. This decoupling makes it easier to develop, scale and run applications. Docker is based on client-server architecture. The ‘Docker daemon’ does the heavy lifting tasks of building, deploying and managing objects. Whereas the ‘Docker client’ is a command line interface that interacts with the Docker daemon.

Before we get started I would like to introduce a few terms that we will using in this article on Docker:

- Image: Read-only template with intructions to create a Docker container. Information stored in image is in JSON format.
- Container: A runnable instance of an image. It is configurable and isolated from other containers.
- Service: allows you to define the expect state of an application.
- Dockerfile: A textfile with instructions to create an image.
- Docker Store: A public repository of Docker images

First step in using Docker is to pull a base version of an image and then install the applications file and libraries on top of this base image. We can perform this process manually but it as complexity grows it is quite likely that this method would become impossible to manage. A much better alternative is to use Dockerfile. A Dockerfile is a series of intructions to create an image. It is present in the root folder of the application. This Dockerfile is consumed by build command to produce a custom image of application.

The simplest Dockerfile will perform the following operations:

1. Pull base image of OS from Docker Store
2. Install libraries and binaries

3. Copy application files to container
4. Set working directory
5. Set of commands to run when container starts

The build process for creating image will consume Dockerfile and produce our image. This image is then used to run container that holds our application.

```
docker image build -t [container-name] .
docker container run [container-name]
```

An image is not a single monolithic block. It is composed of several layers, with each Dockerfile instruction generally representing a separate layer. Each layer is immutable and builds on top of previous layers. A dedicated layer keeps tracks of changes being made to an image. So, when we rebuild an image, the process does not start from scratch. Instead, Docker intelligently builds new image from previous builds by utilizing cache. This reconciliation method makes build process extremely fast.

Originally published at blog.adityadixit.me.