

Docker

By Tebogo Dube

Introduction - History

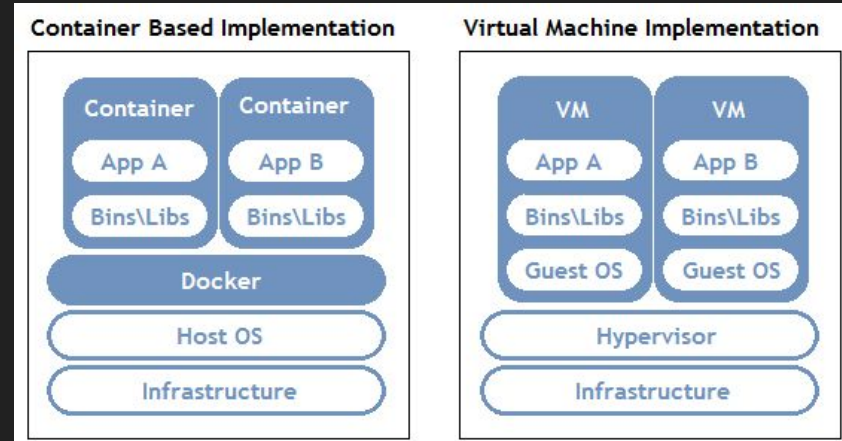
Before Docker and Virtual Machines, if companies wanted to run a specific Server OS they would need to buy a server dedicated to that OS and no other. But then came Hypervisors.

Virtual Machine

Hypervisors allow us to take specified portions of the server to use to run a microserver with an operating system of our choosing, well allowing us to specify the amount of resource and space we want to dedicate to that one specific portion

Docker:

With Docker, it runs on one dedicated OS, with no need for a Hypervisor, the Docker Engine will allow you to create Containers - which are like microserver. This microserver is lightweight compared to a Virtual Machine, meaning it boots up quickly (in seconds) well running any OS, as long as that OS has the same Kernel as the OS the Docker Engine is running on.



Why is it so lightweight and What is a container

The reason Docker Containers are so lightweight is the fact they do need to re-run a Kernel for each microserver they want to run, meaning the Operating system of a Container can just easily communicate with the Kernel of the Host Machine . This also rises the question of size, the microservers of a Docker Container are much smaller as they don't contain a kernel, as well as a Hypervisor

Container

A Container is a set of process that runs in isolation, with no interference or communication of outside process - this means anyone who runs that container, will get the same results as the first time as the container was built. A program that manages these containers, is called a container engine, its job is the communicate with kernel to give the containers access to system resources specified or otherwise.

Docker Engine

With the Docker Engine you can create destroy and manage Containers, like a client server architecture with CRUD functions, but for container management. The Docker daemon is where the Containers are found and run in the background and are controlled by a Docker Client.

Docker Container

A Docker Container consist of application code and other dependencies, needed to build the container. For example a Maven Container, would need maven in its binary files to be able to run.

Docker Image

An image is like a Zip file with the blueprints to create a container, to build an image you would need a Dockerfile in your project, which has instructions on how to make the image, and run it.

Building an Image - Dockerfiles

To create a dockerfile, first we start with the base image, this is the foundation of the image. - In itself, it is a official image provide by docker and it can contain a specified operating system. You can create your own

After the base image, the middle section of the file is usual for setting environments, or copying workdir to the base image.

RUN - this instruction is used when you want to execute a command, while building the image, and only when building the image

CMD - This instruction servers as a way to execute your image, and is ran when your container starts up

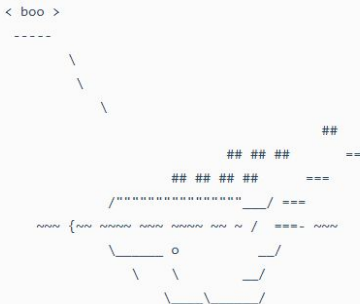
```
1 # we will use openjdk 8 with alpine as it is a very small linux distro
2 FROM openjdk:8-jre-alpine3.9
3
4 # copy the packaged jar file into our docker image
5 COPY target/demo-0.0.1-SNAPSHOT.jar /demo.jar
6
7 # set the startup command to execute the jar
8 CMD ["java", "-jar", "/demo.jar"]
```

Docker Hub

Docker Hub is a repository with a vast amount of Images, that are shared with other developers. You can find pre-build images here to use and manipulate to your liking, most images on docker hub come with detailed instructions on how to use them and set environments if needed. If I wanted an image from Docker hub all I would need to do in my command line is run `docker pull` and the name of the Image I would like

How to use this image

```
$ docker run docker/whalesay cowsay boo
```




```
tebogo@ubuntu:~$ docker pull docker/whalesay
```

Using default tag: latest
latest: Pulling from docker/whalesay
Image docker.io/docker/whalesay:latest uses out of date schema2 image for better future upgrade to a schema2 image for better future at <https://docs.docker.com/registry/spec/daemon/>
e190868d63f8: Extracting 63.5MB/65.77MB
909cd34c6fd7: Download complete
0b9bfabab7c1: Download complete
a3ed95caeb02: Download complete
00bf65475aba: Download complete
c57b6bcc83e3: Download complete
8978f6879e2f: Download complete
8eed3712d2cf: Download complete

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
hello-world	latest	bf756fb1ae65	11 months ago	13.3kB
webcenter/activemq	5.14.3	ab2a33f6de2b	3 years ago	422MB
docker/whalesay	latest	6b362a9f73eb	5 years ago	247MB

```
tebogo@ubuntu:~$ docker run docker/whalesay cowsay Hello-World
```



Reference

- Docker Documentation. 2020. *Docker Hub Quickstart*. [online] Available at: <<https://docs.docker.com/docker-hub/#:~:text=Docker%20Hub%20is%20a%20service,container%20images%20with%20your%20team.&text=Builds%3A%20Automatically%20build%20container%20images,Docker%20Hub%20with%20other%20services.>> [Accessed 17 December 2020].
- Hiwarale, U., 2018. *Anatomy Of Docker*. [online] Medium. Available at: <<https://medium.com/sysf/getting-started-with-docker-1-b4dc83e64389>> [Accessed 24 November 2018].
- Hub.docker.com. 2015. *Docker/Whalesay*. [online] Available at: <<https://hub.docker.com/r/docker/whalesay/>> [Accessed 17 December 2020].
- Kapeli.com. 2020. *Dockerfile*. [online] Available at: <https://codefresh.io/docker-tutorial/create-docker-images-for-java/https://kapeli.com/cheat_sheets/Dockerfile.docset/Contents/Resources/Documents/index> [Accessed 17 December 2020].
- NetworkChuck, 2020. *You Need To Learn Docker RIGHT NOW!! // Docker Containers 101*. [video] Available at: <https://www.youtube.com/watch?v=eGz9DS-aleY&ab_channel=NetworkChuck> [Accessed 1 April 2020].

Reference

References

2020. *Docker Architecture*. [image] Available at:

<<https://www.aquasec.com/cloud-native-academy/docker-container/docker-architecture/>> [Accessed 17 December 2020].

Baker, A., 2020. *Three Ways To Create Docker Images For Java - Codefresh*. [online] Codefresh. Available at:

<<https://codefresh.io/docker-tutorial/create-docker-images-for-java/>> [Accessed 5 March 2020].

Davis, M., 2017. *Dedicated Servers Vs Virtual Servers: What's The Difference? (Updated)*. [online] Future Hosting. Available at:

<<https://www.futurehosting.com/blog/dedicated-servers-vs-virtual-servers-whats-the-difference/>> [Accessed 6 November 2017].