

Docker

- ⇒ Containers → A way to package application with all the necessary dependencies and configuration
- Portable, can be easily shared and moved around.

→ Docker hub

Application Development

Before Containers

- Applications required every developer had had to install services and dependencies in their local environment and then configure them and run them in local operating system.
- Required many steps where things could go wrong.

After Containers

- Doesn't have to install services and dependencies directly on your operating system.
- Containers is ^{its} own OS layer, package with all the configuration
- checkout the container, download on your local machine and start the project
- one command to install the app.

Application Deployment :-

Before Containers

- configuration on the server needed.
- dependencies version conflicts.
- textual guide of deployment.
- misunderstandings

After Containers

- Developers and operations work together to package the application in a container.
- No environmental configurations needed on server, except ^{Docker runtime}
- only use docker command to pull the docker image from containers repo and run it.

→ Docker Image vs Docker Container.

- ↳ actual package
- ↳ artifact, that can be moved around.



not running

actually start the application
(i.e. pull the image on local machine and starting it).

↳ container environment is created



running

→ You can run multiple number of same applications with different version with no problem at all.
Both are virtualization tools.

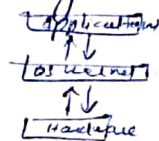
→ Docker vs Virtual Machine

↳ operating system has two layers

↳ 1) OS kernel

↳ 2) Applications

↳ e.g. different linux distribution
↳ same kernel
↳ different applications (GUI)



↳ It has both application and OS kernel layer, hence virtualizes the complete operating system.

Docker virtualizes the applications layer and uses the same kernel of host.

→ So when pull docker images it actually contains applications layer of operating system and have some applications installed in top of it and uses the kernel of the host as it doesn't have its own kernel.

→ Docker size is smaller (in MBs)

→ starts and run much faster

→ Docker image have compatibility issues with OS host.

→ So when you download virtual machine image on your host, it doesn't use host instead boot up own host.

→ VM size is larger (in GBs)

→ run slower

→ VM of any OS can run any OS host

QA Test Environments → Quality Assurance Environments
↳ set up to perform software testing for validation of its working.

- Containers make development and deployment more easy and efficient.
- Containerizing the applications. ⇒ Dockerizing the application.
- Container is nothing but a combination of layers of images combined running in a environment.

