

Module - 3

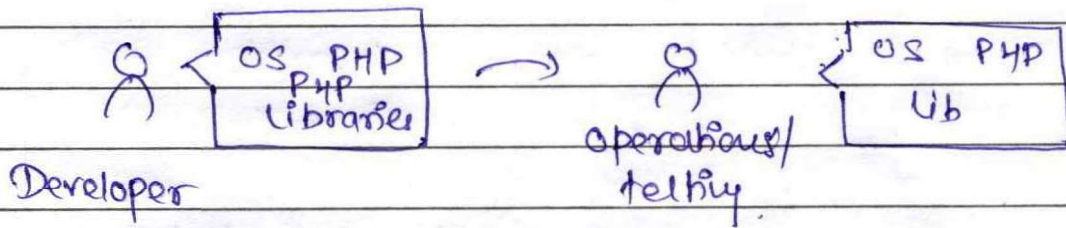
Date ___/___/___

Page ___

Containerization Using Docker

* Problems before Docker :-

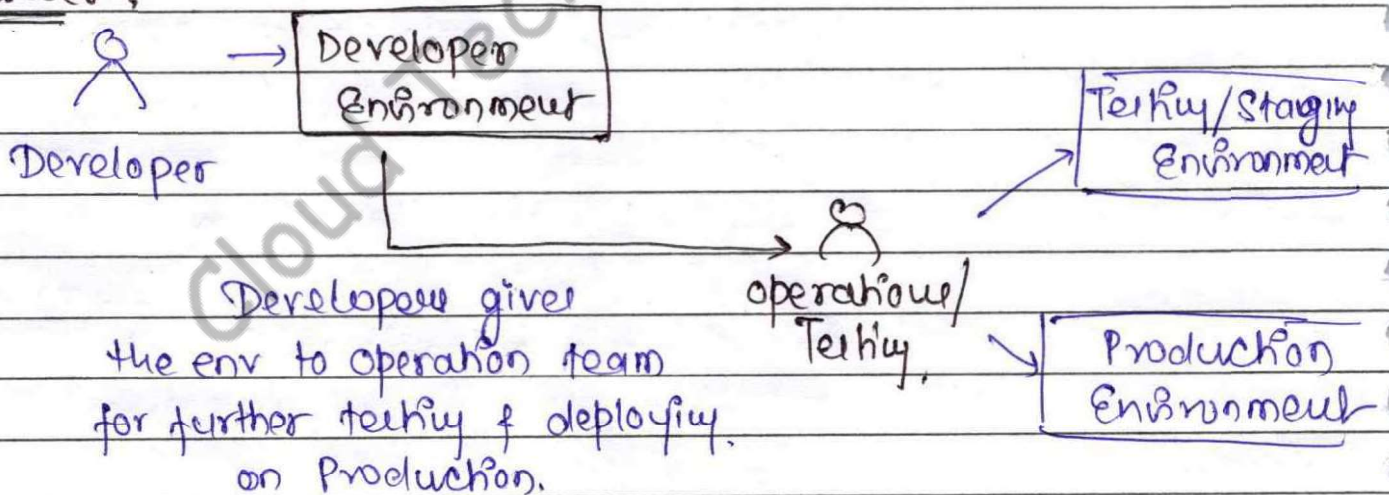
Imagine you are a developer, and you are creating a website on PHP (contains OS, PHP software, libraries).

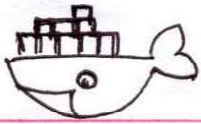


Developers used to run the code on their system, it would run perfectly, But the same code did not run on the operations team system.

The Problem :- we needed an entity which can "contain" all the software dependencies, and can be ported on to the other computers as plug & play package.

Answer :-





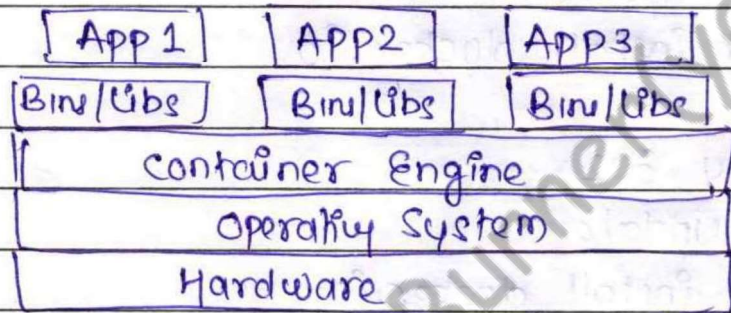
A wrapper

Date ___/___/___

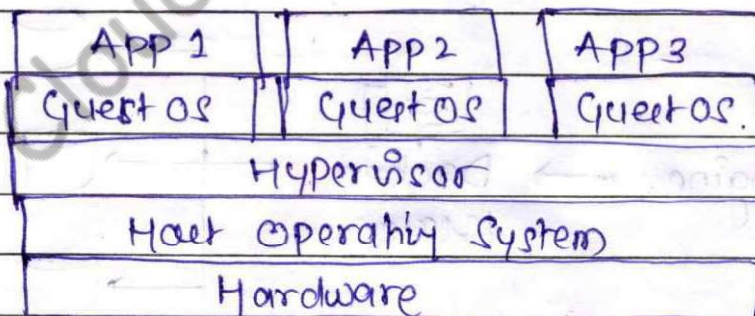
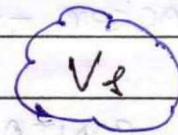
Page 7

DOCKER :- Docker is a computer program that performs operating-system-level virtualization, also known as "containerization". It was first released in 2013 and is developed by Docker, Inc. Docker is used to run software packages called "containers".

↳ Application containerization is an OS-level virtualization method used to deploy and run distributed applications without launching an entire virtual machine (VM) for each app



Docker
Architecture



Virtual M/c
Architecture

DOCKER INSTALLATION :-

↳ Installation on Windows :-

One needs to install Docker Toolbox for windows, for using Docker

"<https://store.docker.com/editions/community/docker-ce-desktop-windows>"

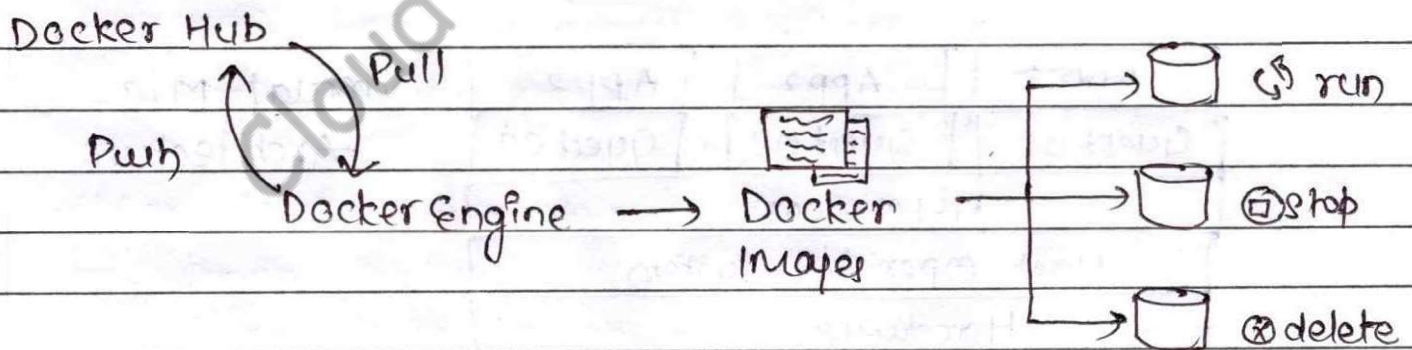
↳ Installation on Ubuntu :-

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

Now run on Ubuntu EC2

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

* DOCKER CONTAINER LIFECYCLE



* COMMON DOCKER OPERATIONS :-

1. ^{sudo} docker pull ubuntu ^{image name}
This command helps you pull images from the central docker repo.
2. sudo docker images
↳ It helps you in listing all the docker images, downloaded on your system
3. sudo docker run -it -d ubuntu
↳ to run the container from their image name
^{interactive mode (so we can pass other commands)}
^{daemon (it should run in background)}
4. sudo docker ps
↳ List of all "running" containers
5. sudo docker ps -a
↳ List of all container either running or stopped
6. sudo docker exec -it <container ID> bash
↳ to go inside the container & act as root.
^{to run container in current terminal}
- sudo apt-get update
7. sudo docker stop <container ID>
↳ to stop the container
8. sudo docker kill <container ID>
↳ to kill it instant

9. `sudo docker rm <container ID>`

↳ To remove a stopped container from the system.

10. `sudo docker rmi <image-id>`

↳ To remove an image from system.

* CREATING DOCKER HUB ACCOUNT :-

↳ go to hub.docker.com

↳ enter unique docker ID & password to signup

* SAVE CHANGES TO A CONTAINER

- `sudo docker run -it -d ubuntu`

- `sudo docker ps`

- `sudo docker exec -it <container ID> bash`

- `ll`

- `mkdir app` (to create a directory)

- `sudo docker rm -f <container ID>`

after delete and when you do login to container

- `sudo docker run -it -d ubuntu`

- `sudo docker exec -it <container id> bash`

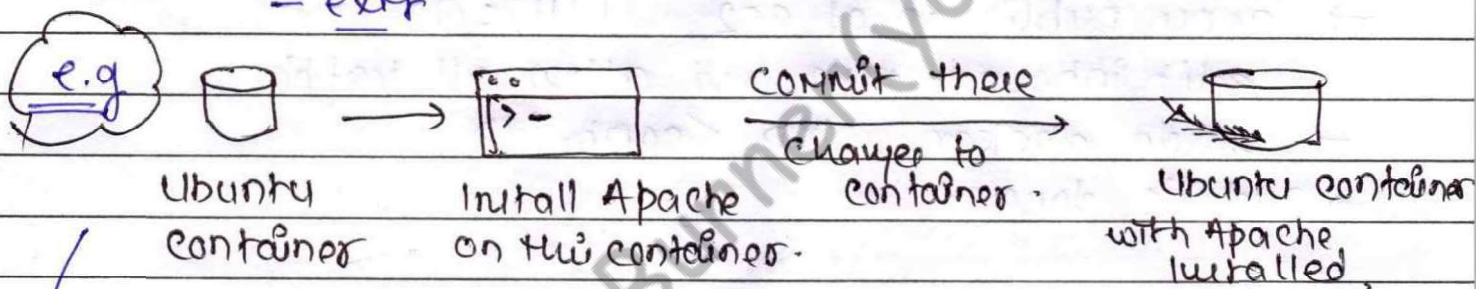
you will did not find that directory (app) which you created earlier.

To save there type of changes.

1. `docker commit <container-id> <name-for-image>`
 ↳ To commit the changes and give your name for this new image

- `sudo mkdir app`
- `ls`
- `exit`
- `sudo docker ps`
- `sudo docker commit <container-id> test`
- `sudo docker images`
- `sudo docker run -it -d test`
- `sudo docker exec -it <cont-id> bash`
- `exit`

name of image



- `sudo docker rm -f $(sudo docker ps -a -q)`
 ↳ to delete all the containers at once.

- `sudo docker run -it -d ubuntu`
- `sudo docker exec -it <cont-id> bash`
- `apt-get update`
- `apt-get install apache2`
- `service apache2 status`
- `service apache2 start`
- `service apache2 status`
- `exit`

→ `sudo docker ps`

→ `sudo docker commit <cont-id> cloudtech/apache`