**Lecture 1:**

Book: Docker Deep Dive by Nigel Poulton

Old Days:

1 app – 1 server: Because

1. Lack of technical knowledge for resource allocation understanding. / slow hardware.

2. Different infrastructure and dependencies.

Disadvantages:

1. Very costly

2. Resource wastage

3. Many servers to manage.

1998 VMware company: Built applications for virtualization

\* Built VMware workstation, first product, 1999

Benefits :

1. Multiple apps, single server

2. Different OS and dependencies on same server

3. Much better then old one

4. Saves a lot of resources.

Disadvantages

1. OS consumes a lot of resources

2. Licensing cost for every OS instance.

Hypervisor:

a layer between guest OSs and base/hose OS. Also called… , isolates operating systems from hardware.

This allowes the host machine hardware to independently operate one or more virtual guest machines.

**Lecture 2:**

Containers: We use them to run independent, isolated applications. Each container is an isolated region which keeps different container applications to interact with each other as well as provide a separate environment for each.

Google has been using this tech for a long time.

Infrastructure-→ Operating System → Docker engine → App1 || App2 || App3 (each app in a separate container.

Single OS and so, less cost (license ) less RAM and processing need,

Containers are very protable. Can run anywhere

Containers are fast.

Once an application works, it will work everywhere, because the said environment will be created before installing the application on anyserver.

Cap-ex and Op-ex are reduced. (less licensing fees and less servers needed, i.e. less human resource needed.)

**Lecture 3:**

Types:

1. Windows Containers: Called windows containers, made with the collaboration between microsoft and dockers.

2. Linux Containers: Nothing new, made by the contribution of many free contriburors, including Google.

3. MacContainer: No MACcontainers. But Linux containers can be used on MAC

Windows containers cannot run on linux containers. BUT… Dockers containers can work everywhere.

**Lecture 4:**

Docker is a software that manages and provides services for containers. Its open source and is placed on github.

San Fransisco based. Without Docker engine, we wouldn’t be able to use containers this easily.

Dockers available as :

1. Enterprise edition EE (paid)

2. Community edition CE

**Lecture 5:**

Terms use of included but removable batteries. Means we can replace built in provided stacks with any 3rd party stack. (example networking)

OCI (Open container initiative.) : Governing body, joined with linux foundation and made some standards for image formats and containers run-times.

**Lecture 6:**

Docker installation:

docker - - version

docker version

docker info

**Lecture 7**

Docker engine is core software used for making and managing containers.

Docker client – used to communicate with docker deamon and docker server

Containerd, runc etc are part of this structure.

Docker deamon- listens to docker api and helps manage docker objects (images and containers)

Containerd – bridge between deamon and runc (pronounced as run c) Makes, destroys, pauses, resumes containers etc.

runc – often named as runtime, is used to make a container.

Shim - shim component is just between containerd (pronounced as container d) and runc. It makes any container a deamonless container. You can upgrade docker deamon and the working containers will keep on working.

**Lecture 8**

Images – stand alone, lightweight, executable software package. Includes code, runtime, system tools, system libraries, settings. All necessities for running the software.

Images, when run on docker engine, becomes containers.

Images are made up of layesr on top of each other and represents a single object.

Images can be made containers and containers can be made images.

Image dependencies are scanned and if some dependency that was previously used by some other container is found, it is not downloaded. The rest are acquired.

TO delete and image, you need to delete all the containers you have made from it. Even if one container lives, the image cannot be deleted.

Alpine linux Docker image size is 4mb and ubuntu docker image is 120mb. They are stripped off non essential items.

**Lecture 9:**

Image registeries:

Docker images are portable. i.e. they can be saved on any registery or server.

Docker-Hub: default registery for docker . If you will upload any image, it will be done on docker hub

You can save every version of a certain image.

Docker hub provides official repositories to companies. Free repositories are called unofficial. (we would be using these)

Make Docker-Hub account

**Lecture 10:**

Image naming and tagging.

2 ways to download images from docker hub:

1. if for an official image (like alpine, ubuntu)

docker image pull <repository name>:<tag>

e.g: docker image pull ubuntu:18.04 ( will download the image of ubuntu 18.04)

2. for unofficial images:

docker image pull <username>/<repository name>:<tag>

If tag is not given, docker will download the image tagged with “latest” word . ( It is not necessory that an image will be latest if it is tagged as such)

Same goes for push. (upload)

Lets make a small image.

docker image pull alpine:latest