// Docker and Kubernatics Course:

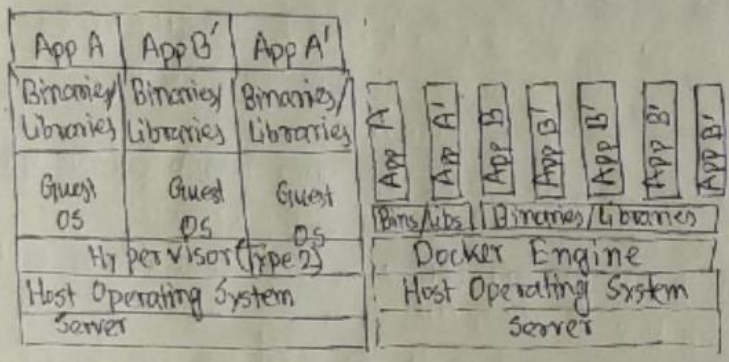
Date: 17 July 2020

Tecognize Training: (Class-01: 17 July 2020)

// Task: HackerRank এর maximum problem (python related) সমাধান করতে হবে।

Docker and Virtual Machine এর পার্থক্য ?

// REST API বানানো শিখতে হবে।
// programming language শিখতে হবে যেকোন একটা।
//



Docker এ guest OS থাকে না কেন ?

1. Docker uses a client-server architecture
2. Docker Clients talks to the docker deamon, which does the heavy lifting.

HackerRank
↓
// Python exercise গুলো করে ফেলতে হবে //

docker ps
docker pull mongo:latest
↓
এই command গুলো Docker server কে দিয়ে execute করাতে।

// github.com open করেছে।

// এখন থেকে note গুলো রাখবে & share করতে হবে।

// First week বেশি effort দিবে।

// python এ try দিচ্ছে hard and first.

// HackerRank এ problem fixed করা হয়, যত বেশি possible.

// "mark down file" শিখেছি।

☐ Git Hub এ markdown file editor "StackEdit". এ গিয়ে ফ্রী ফ্রী লিখবো regular basis এ share করবে।

☐ Docker Doc : Documentation

* Representational State of Application (REST API)
  * Study how REST API works.
  * GET, POST গুলো HTTP request. আমার current state টাকে server কে http এর through এ represent করলে, এটাই REST API.

Docker Architecture:

① Docker Images ② Docker Registries ③ Docker Containers

// Docker Container এর ভেতর একটা process চলে।

// Linux kernel দিয়ে container তৈরি হয়েছে। coz, kernel এর feature দিয়ে আলাদা docker বানায়।

// Snapshot হলো binary image of a file system.

// persistancy = docker দিয়ে database চালানো যায় না।
docker is affemoral; (Affimeral means, process যখন run হয়, ঐ process টা affimeral হয়, process শেষ হয়ে গেলে ঐ process এর সব temporary, process শেষ হয়ে যায়। তাই docker দিয়ে database চালানো যায় না।

Solution হল:

// Image is a snapshot
// container হল ঐ image যে use করে process run হয়

Container আর Image এর মধ্যে পার্থক্য:

⇒ process যখন run হয় তার আগে সেটা file থাকে।

→ run হলে RAM এ load হয়, then CPU clock দিয়ে কাজ শুরু হবে

= Docker Image হল ঐ file টা (যেটা run হয়নি)

- Docker container হল ঐ process টা।

ECR (Elastic Container Registry)

→ Docker image বানাব। then pull করব to docker registry. (public/private)

⇒ public registry তে pull করা উচিত না বিপদ হতে পারে। কেন, server info, password, port এসব info থাকে image এ.

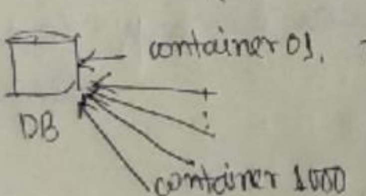
Go-lang .JAR, .WAR, .XML

- Docker install করতে হবে computer এ
- System Design কাজ হল একটা devops engineer এর কাজ।

//

Kubernatics হল workstation: হাজার হাজার container generate করে দেয়। ধরা যাক যে database এ load পড়ল, database crash করল। So, আমাদের dev. and ops দুই world কে মিলিয়ে connect করতে হবে।

Kubernatics - container কে scale up, মানে container এর count বাড়িয়ে দেয়।



যখন 300 container launge হবে তখন কি করতে হবে?

Kubernetes কেমন scaleup করবে তার impact টা আমার দেখাতে হবে,

(Kernel → socket → Application → Database এর মাধ্যমে)

// Docker Images :
// Docker Registries : (Docker registries hold docker images)
// Docker Containers :

Image বানানো :

Docker client → Docker Host এর daemon call করবে → docker

docker file

from ubuntu
from nginx
run nginx

from ubuntu → docker hub এ ubuntu এর image আছে, ঐটা নামাবে।
from nginx → now nginx এর docker image বানাবে।

[docker Hub]

ubuntu ⇒ Hash value থাকে
(Tags এর মাধ্যমে)
Digest

Step 04 : Run the image to generate the container

Step 03 :

Step 02 : nginx image নামাবে।

Step 01 : ubuntu OS এর binary থাকবে (Full image না) ← from docker Hub

Step 03 :
docker ঐ দুইটা image নিয়ে একটা new image বানাবে। new hash value দিবে।
~~[illegible]~~