Docker Practical ✓

Kubernetes – Why & Intro

Kubernetes – Few Terms

---

1 Appⁿ ⟶ Multiple Appⁿ

Order Service

Cart Service

Product Service

Authorization Service

In an org there are multiple MS, for these MS all dependencies, OS, and all things will be handled by Docker

$\boxed{\text{Docker}}$

EC2

✓ Product Service ⟶ 10 Servers

✓ User Service ⟶ 8 Servers

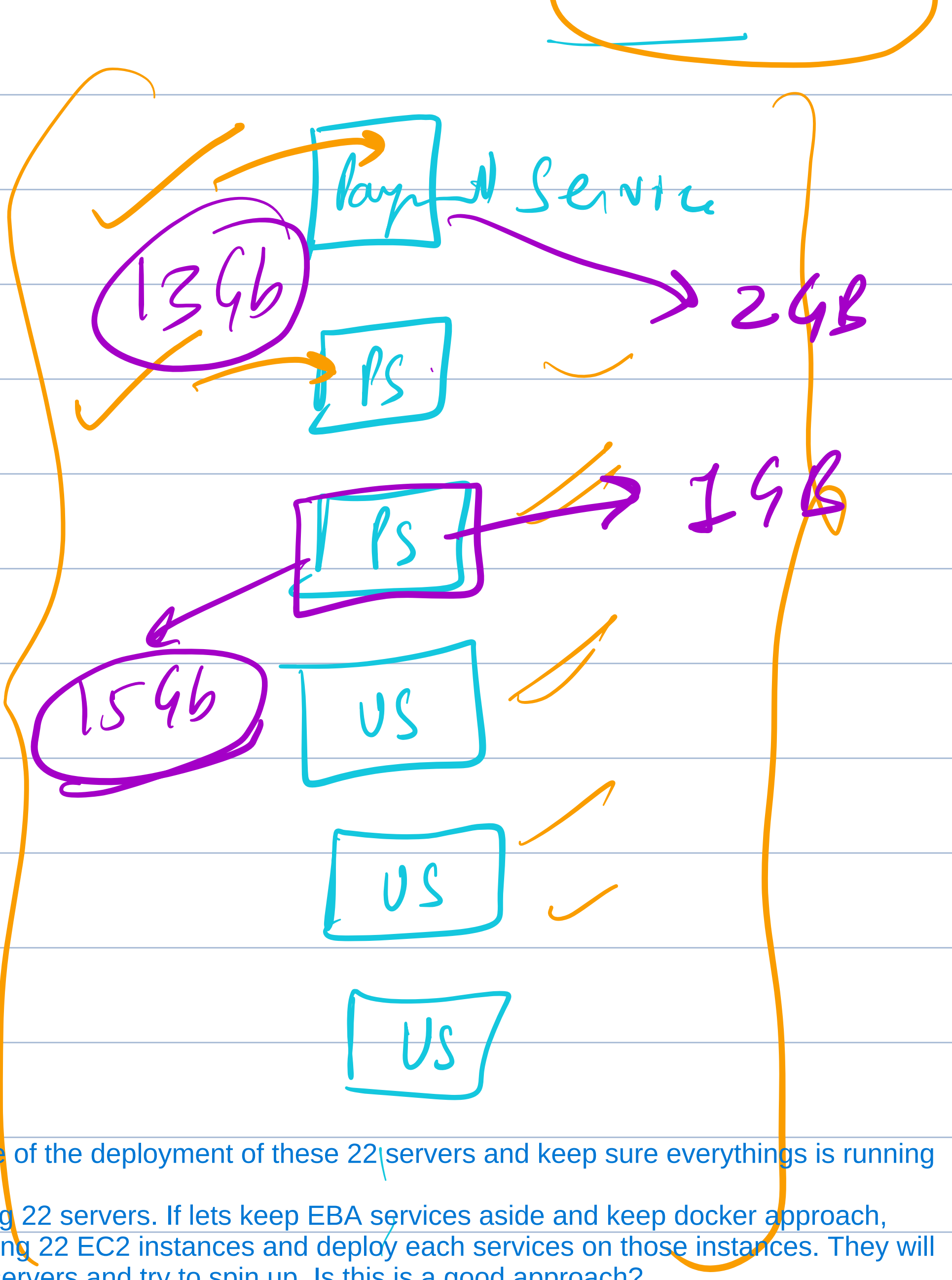
✓ Payment Service ⟶ 4 Servers

Lets say PS require 10 servers to work (handle load since we donot want single point of failure if PS runs on single server) these are the machines running the MSs --> 22 servers.

22 Servers

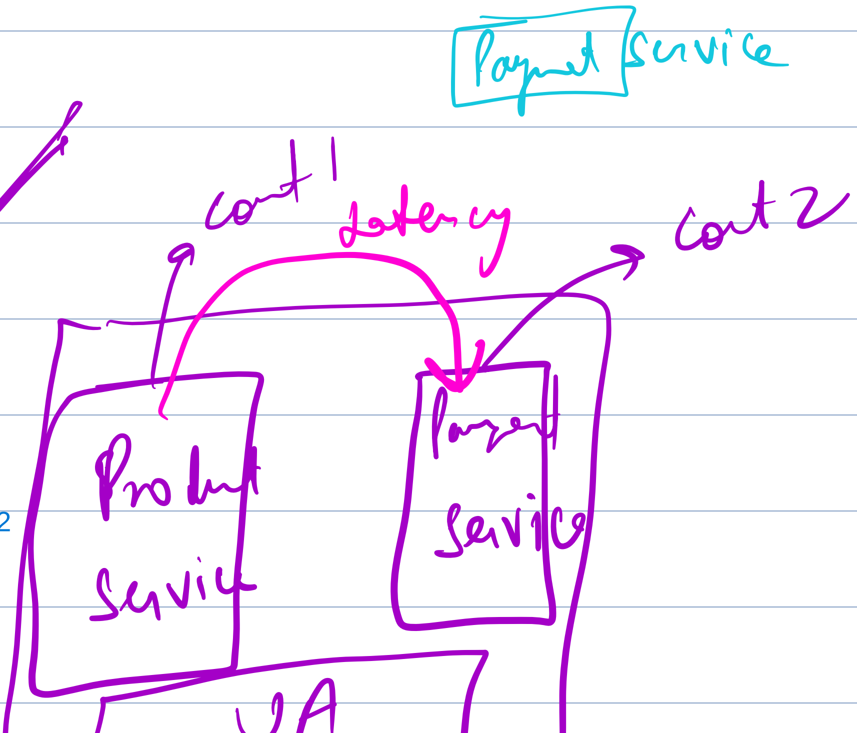
we don't need all servers running all the time

Devops



There will be a DevOps person who will take care of the deployment of these 22 servers and keep sure everythings is running

If there are 22 servers, DevOps will be configuring 22 servers. If lets keep EBA services aside and keep docker approach, what would be the scenario, devops will be creating 22 EC2 instances and deploy each services on those instances. They will upload Docker Image of PS, US, PytS on these servers and try to spin up. Is this is a good approach?



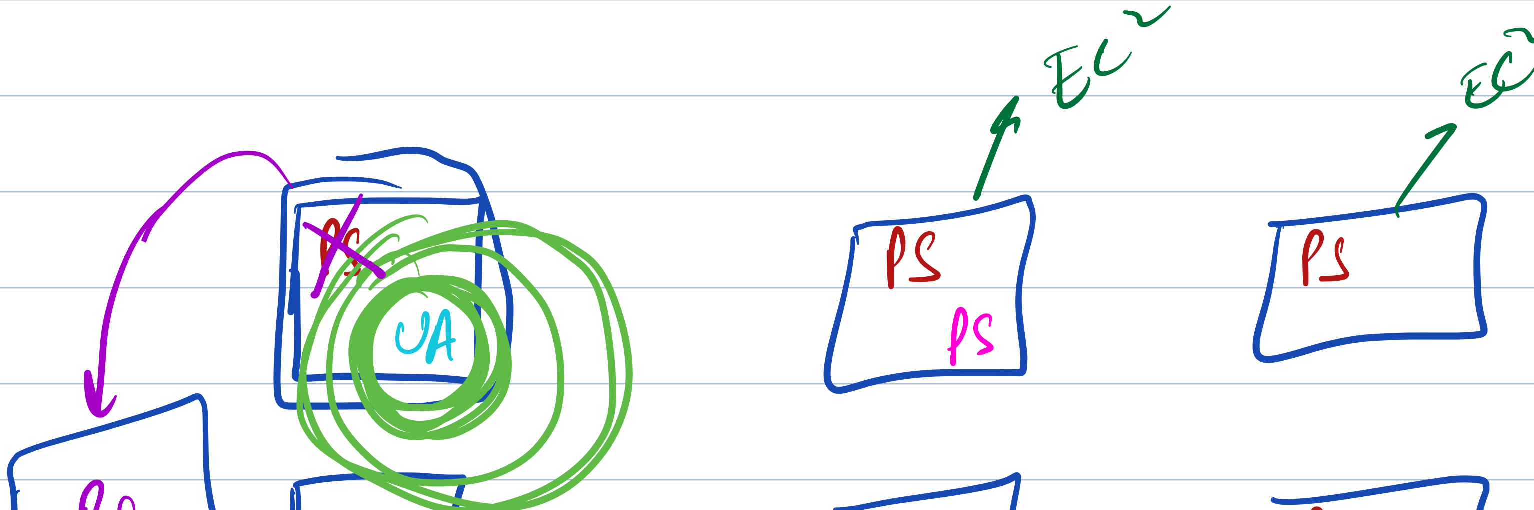
deploy multiple containers in one ec2

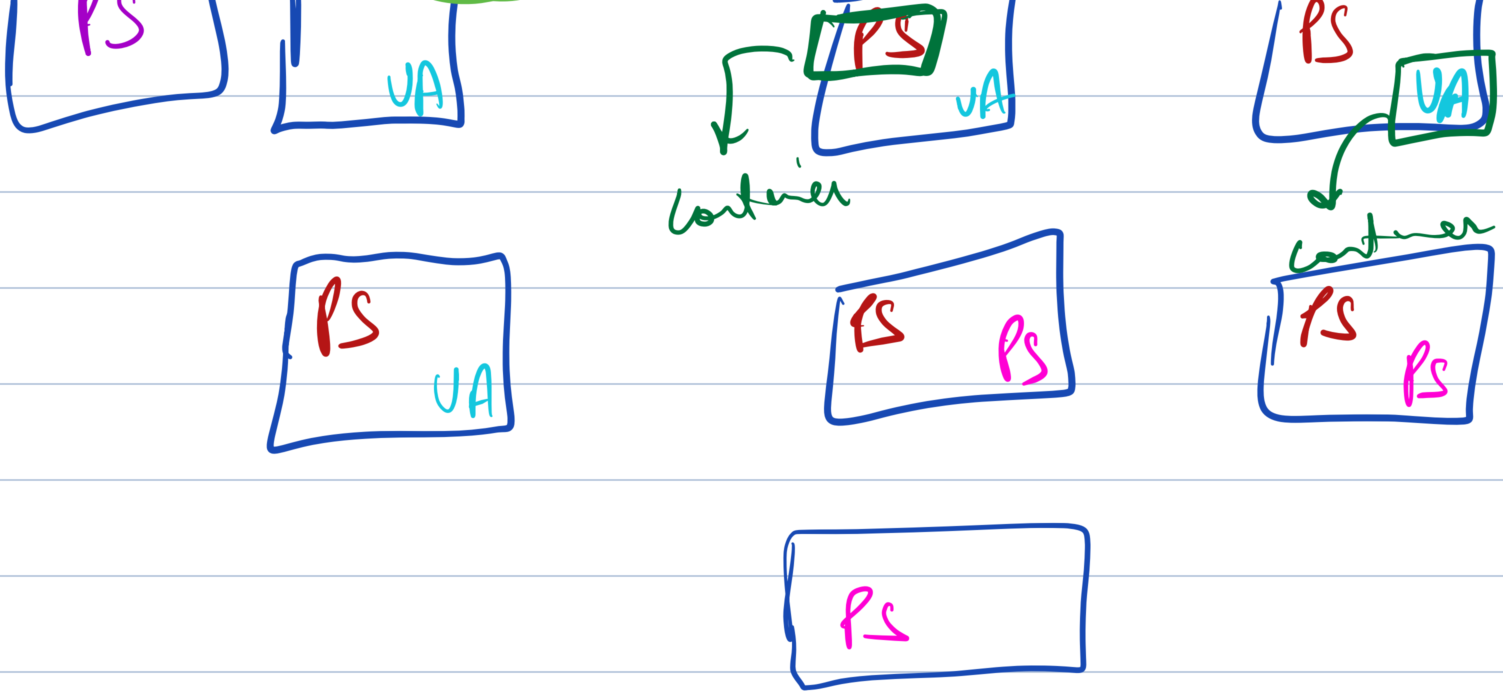
EC2

~~Running~~ only 1 Appn in a server is not an optimal usage of all resources

└ Solution → We can deploy

- multiple containers on one server
- ~~Latency will be~~ very less





Tasks →

Which MS PS + US to be used with which one -> depending on the RAM taken.

✓. Find out Best Arrangement

✓. Deploy as per this Arrangement

✓. Creating a new server (EC^2)

✓. Dynamically moving (EC^2) from one server to other

based on traffic and based of available resources(RAM) in a server --> the arrangement of MS should be changed

• If one server goes down, need to provision one more instance running those containers

We need a orchestartor to handle all of this

Kubernetes

Kubernetes Key Terms →

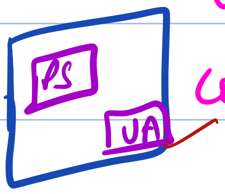
Servers / EC2 / Nodes

just like an employee

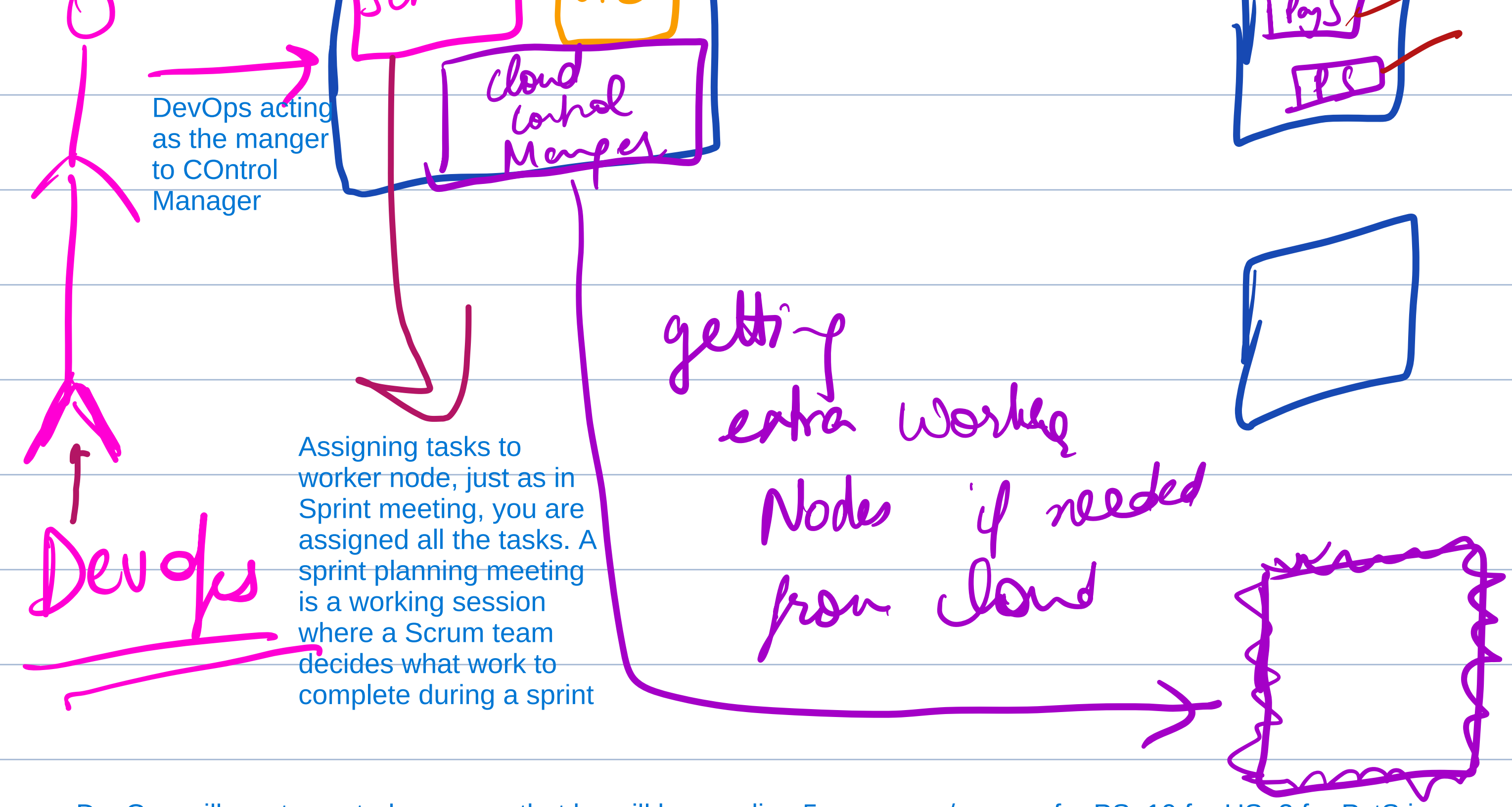
Control Manager just like a manger

[Server that gives instructions / tasks to worker Nodes]

Worker Nodes [Followers who will be following the commands of control Manager]



Scheduler
etcd



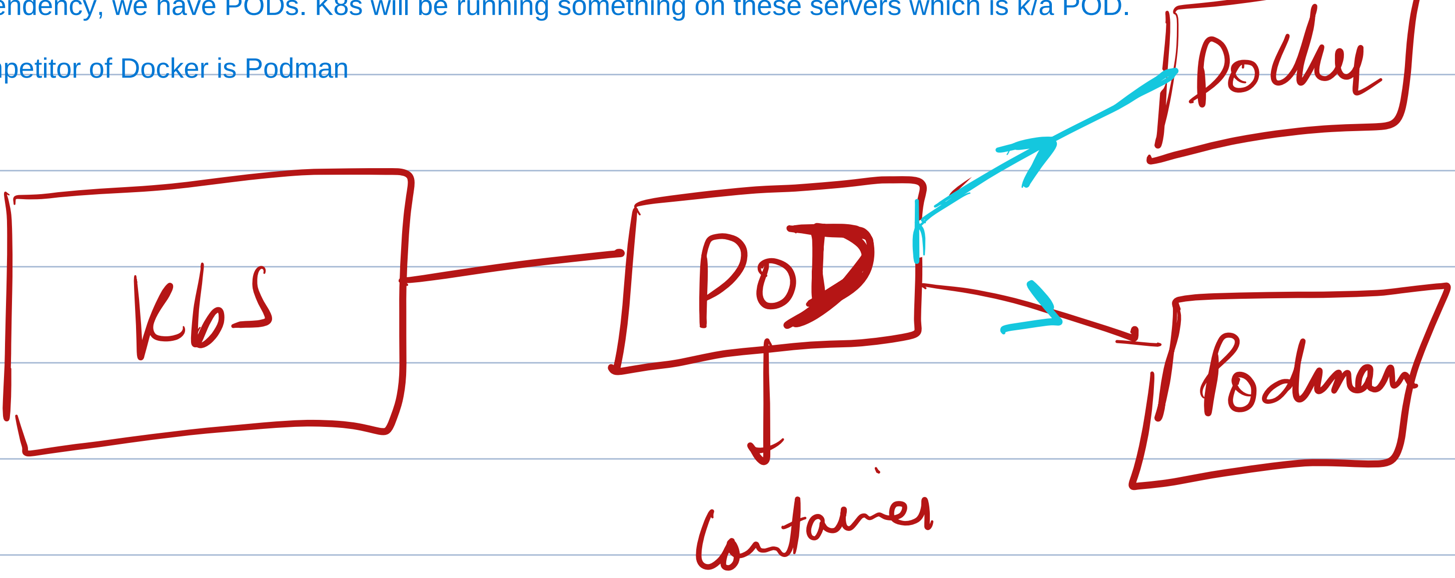
DevOps acting as the manger to COntrol Manager

Assigning tasks to worker node, just as in Sprint meeting, you are assigned all the tasks. A sprint planning meeting is a working session where a Scrum team decides what work to complete during a sprint

DevOps will say to control manager that he will be needing 5 resources/servers for PS, 10 for US, 3 for PytS in the form of config file

At this point of time, in our discussion, K8s had dependency for Docker containers (MS are running on that), to remove dependency, we have PODs. K8s will be running something on these servers which is k/a POD.

Competitor of Docker is Podman



Node
Server

Node is a server

POD → instance of Application

Pod is an instance of application or you can say interface to containerization your application. since interface remove dependency in class (SOLID principles), similar is POD, removing K8s dependency on Docker containers

Interface to containerizing Solution.

DevOps person will give config file to COntrol Manager. there will be 2 config files:-

$\boxed{\text{Deployment Config}}$ → Nuber of worker Nodes which are needed to run particula appn

$\boxed{\text{Service Config}}$ → what will relation of paths to container ports which are running Appn

On one server there will be 2 containers running, and each container can have their own port, PS is running on one port on one container and PytS is running on same port but on another container. But how will external user will know, they want to run something on a port#? If a req came for "/products/1" who will be knowing where to send this req (PS) which node or server and which port I have to transfer it-> Control Manager will be knowing.

Load Balancing

DevOps engineer will give Service config file which will tell which machine is ,what MS, which port to transfer the req n all that

[illegible]