ABB - Openshift

12th June to 14th June 2024

1. Notes
   i. Own Notes
   ii. I also DO create some kind of notes -- Google Drive

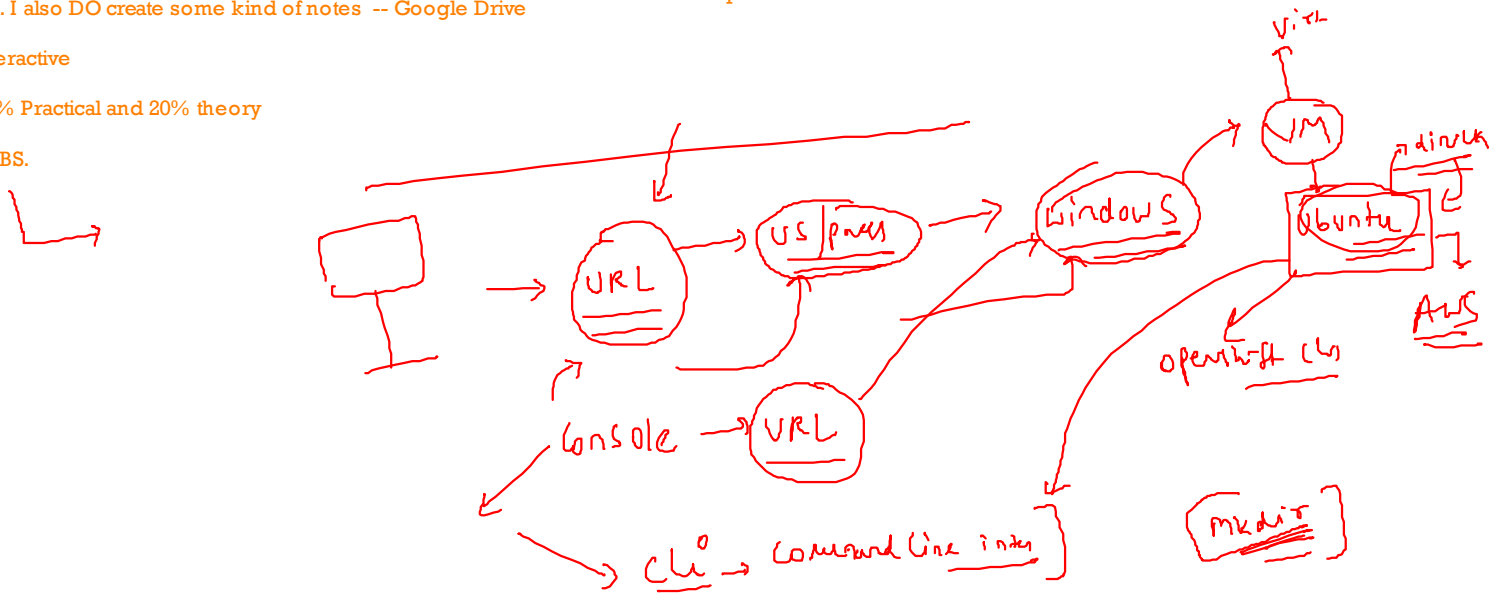
2. Interactive

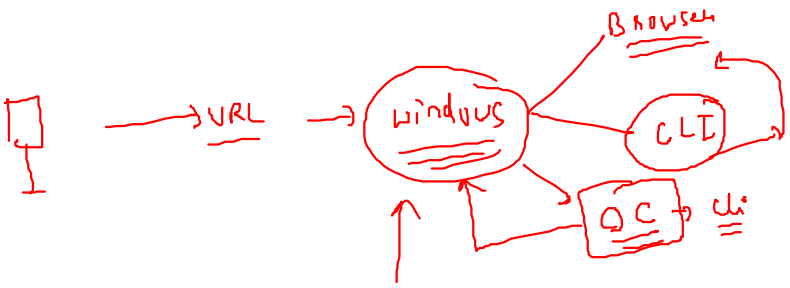
3. 80% Practical and 20% theory

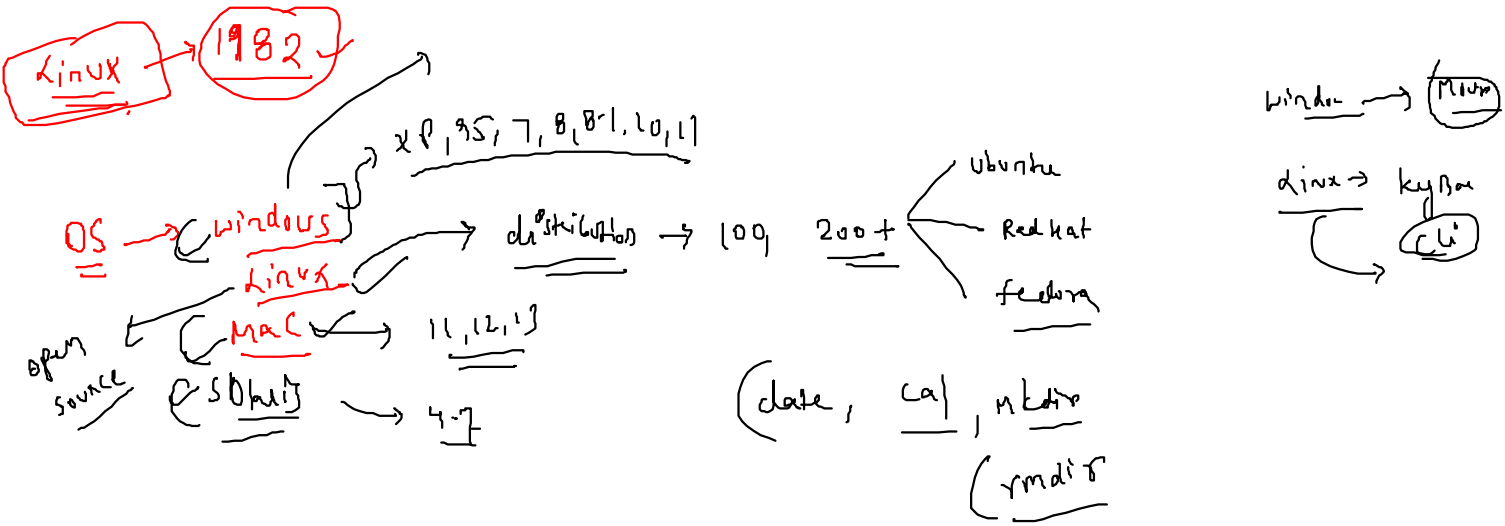
4. LABS.

Agenda :-

1. Linux
2. Docker
3. Kubernetes
4. Openshift
5. Openshift Resources
   a. Projects
   b. Pods
   c. Deployments
   d. Service





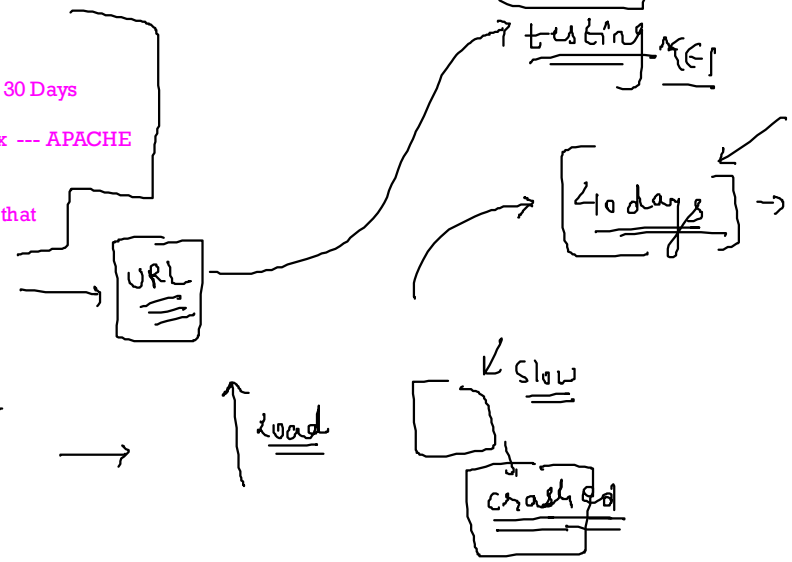


Docker / Kubernetes / Openshift

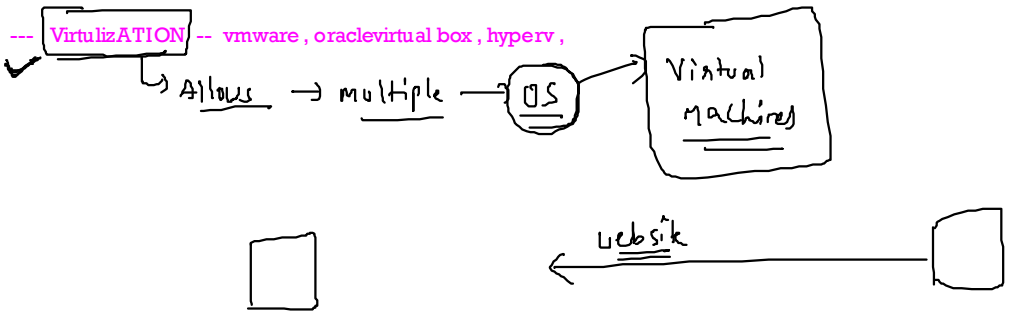
I Phase 1990's --

1. Code -- 45 Days
2. Physical Server --- Order it -- BIG -- Delivery --- 30 Days
3. Install OS -- Windows -- 2 Days
4. INstall Webserver -- Apache , IIS , Tomcat , Nginx --- APACHE
5. COnfigure Apache Server -- 2 Days
6. Test it locally
7. Buy a domain NAme -- connnect our website to that

Total time Taken to create A website == 45 Days

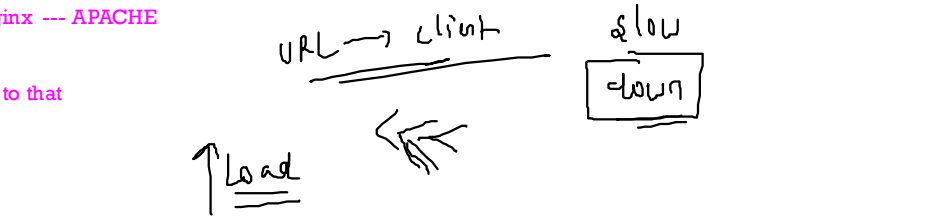


II pHase -- 2000's --- VirtulizATION -- vmware , oraclevirtual box , hyperv ,

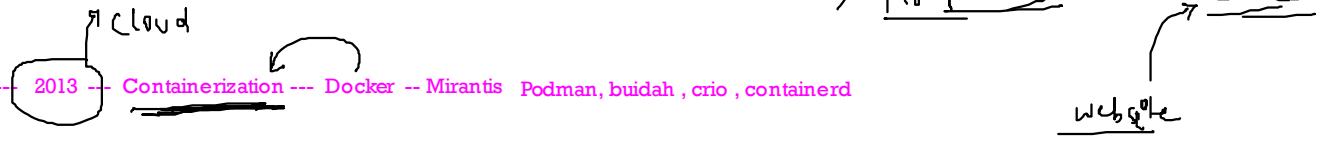


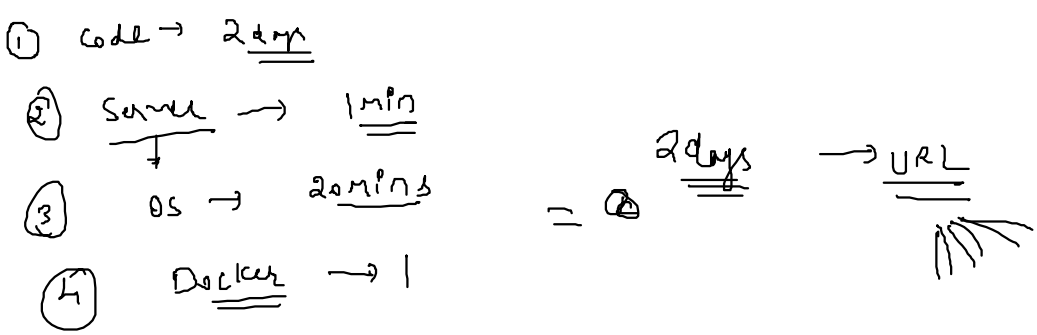
1. Code -- 20 Days
2. Physical Server --- Order it -- BIG -- Delivery ---10 Days
3. Install OS -- Windows -- 6hrs
4. Install Virtulization Software -- VMWARE - 6 hrs
5. Install VM - 6 hrs
6. INstall Webserver -- Apache , IIS , Tomcat , Nginx --- APACHE
7. COnfigure Apache Server -- 6 hrs
8. Test it locally
7. Buy a domain NAme -- connnect our website to that

TOTAL TIME TAKEN :- 20 Days



III Phase --- 2013 --- Containerization --- Docker -- Mirantis Podman, buidah , crio , containerd

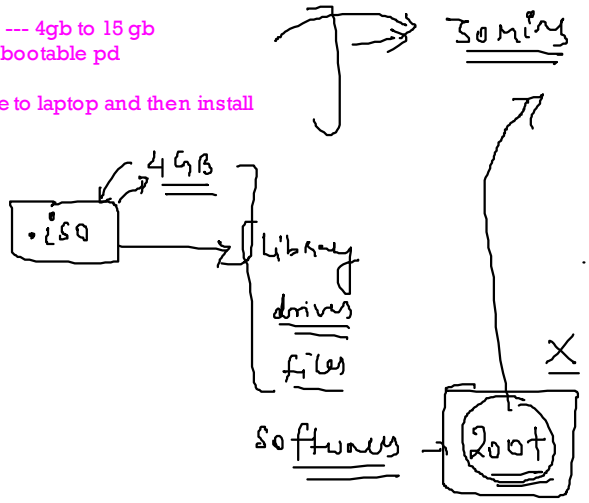


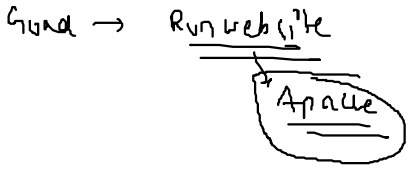


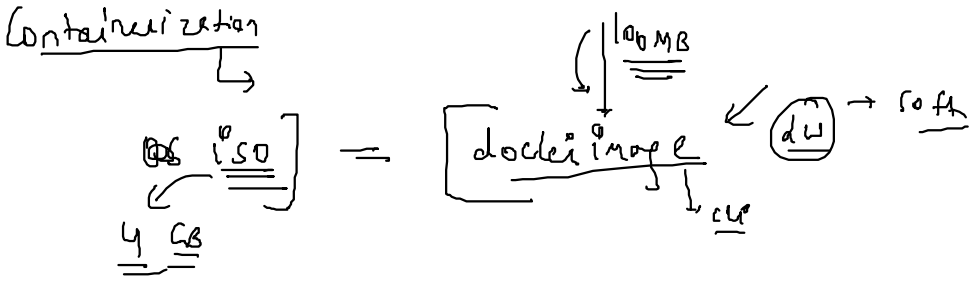
1. Windows --

a. windows.iso --- 4gb to 15 gb
2. OPendrive -- bootable pd

3. attach pendrive to laptop and then install

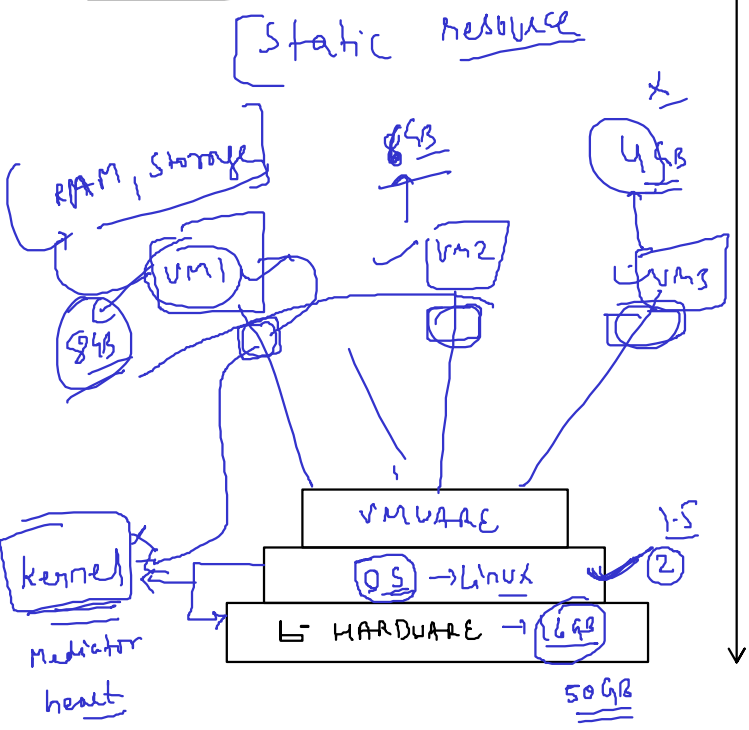




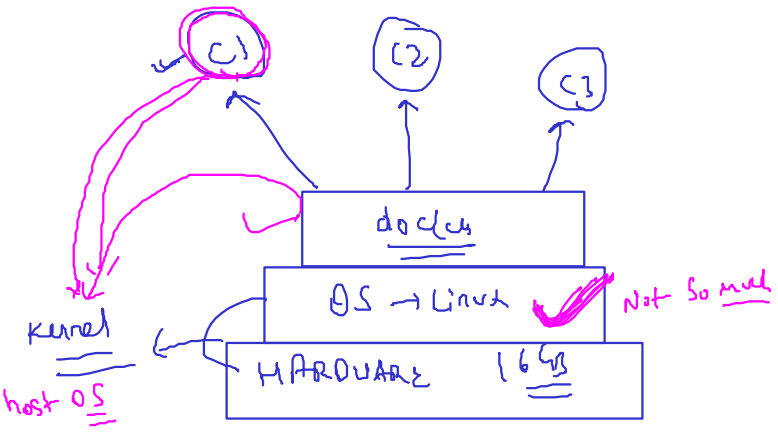


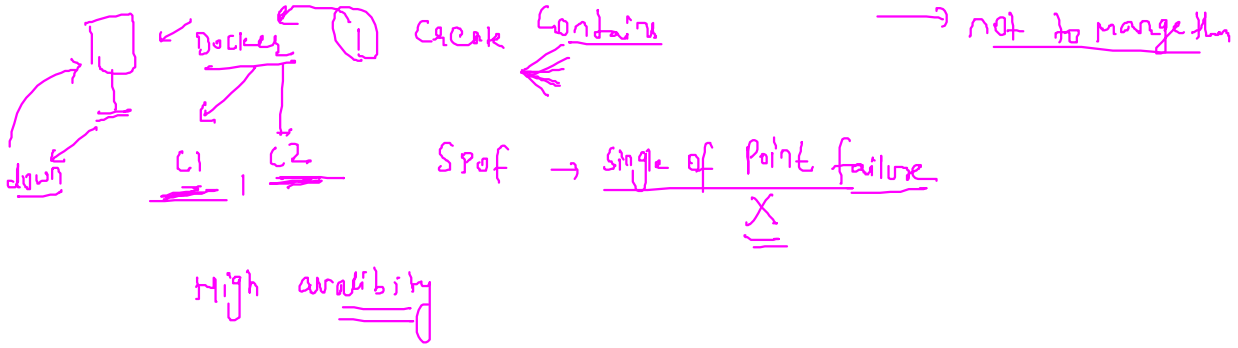
MAIn LAptop	VM	Containerzation
ISO File	ISO file	Docker images
Install OS	INstall VM	Create Containers
		LW os
		light weight os

Virtulization

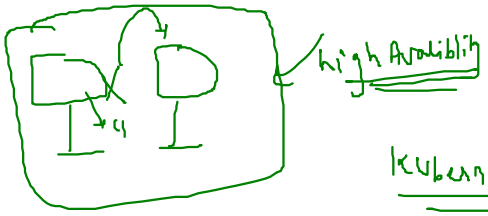


Containerization





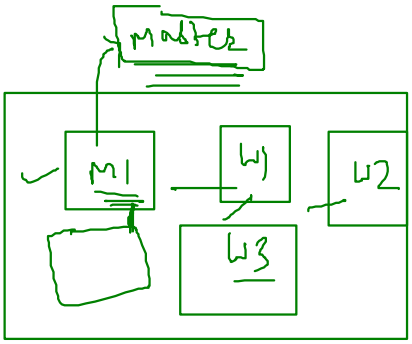
2014 -- Project -- Kubernetes -- Orchestration tool / Mangement Tool for containers --- Google
2016 - Product -- Openshift --- management tool for containers --- Red Hat



clarity → more than 1 m/c

kubernetes → platform → containers

multimaster → high availibility



openshift → cost

Mandatory

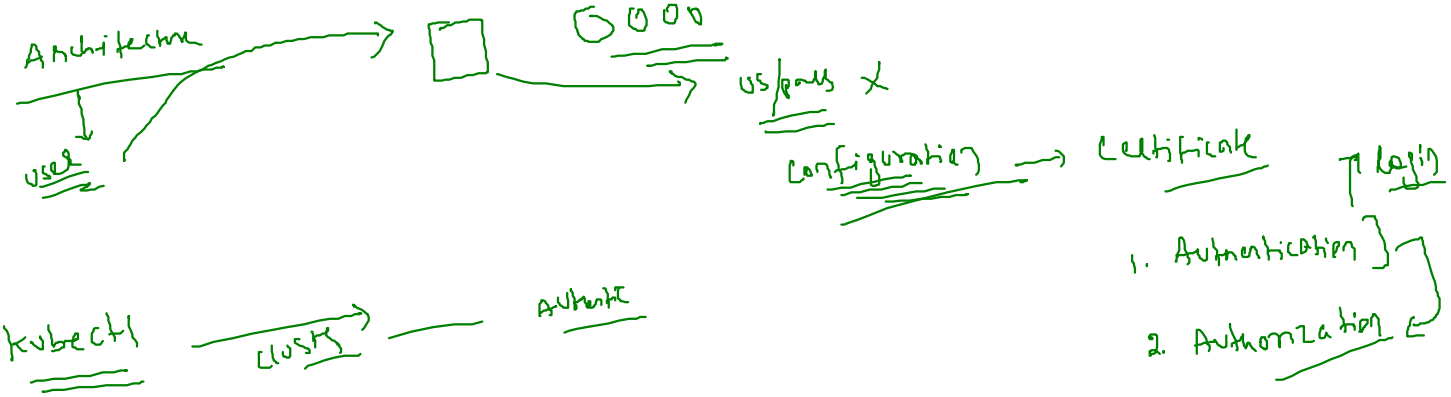
In any cluster of Openshift min 5 machines are req.
3 master machines
2worker machines

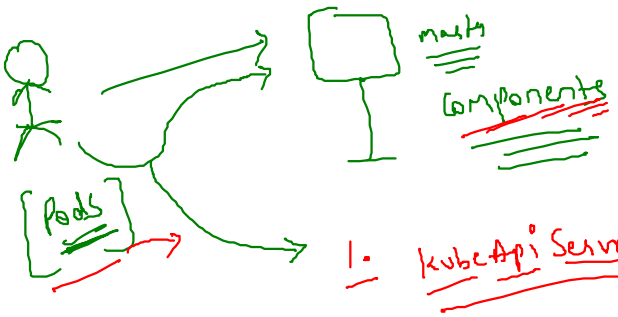
2. OS -- Creating Machines --

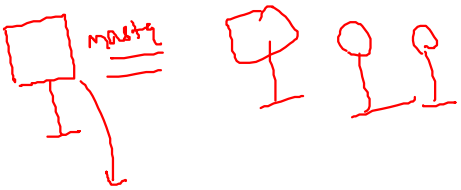
a. RHEL8,RHEL9
b. Red Hat COREOS -- Container Optimized OS

3. EVeyr machine should have min 16gb ram each

80 gb ram -- NO -- AWS CLOUD



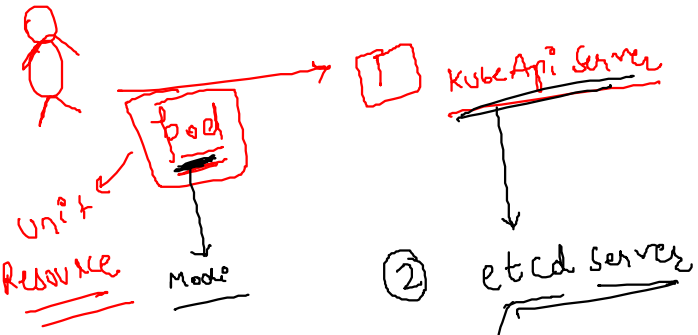




kubectl

pod
Pode

Credentials
Username
Certificate
Configuration



(2) etcd server → datastore key=value

(3) kube scheduler → req. → W1 W2 W3

Resources

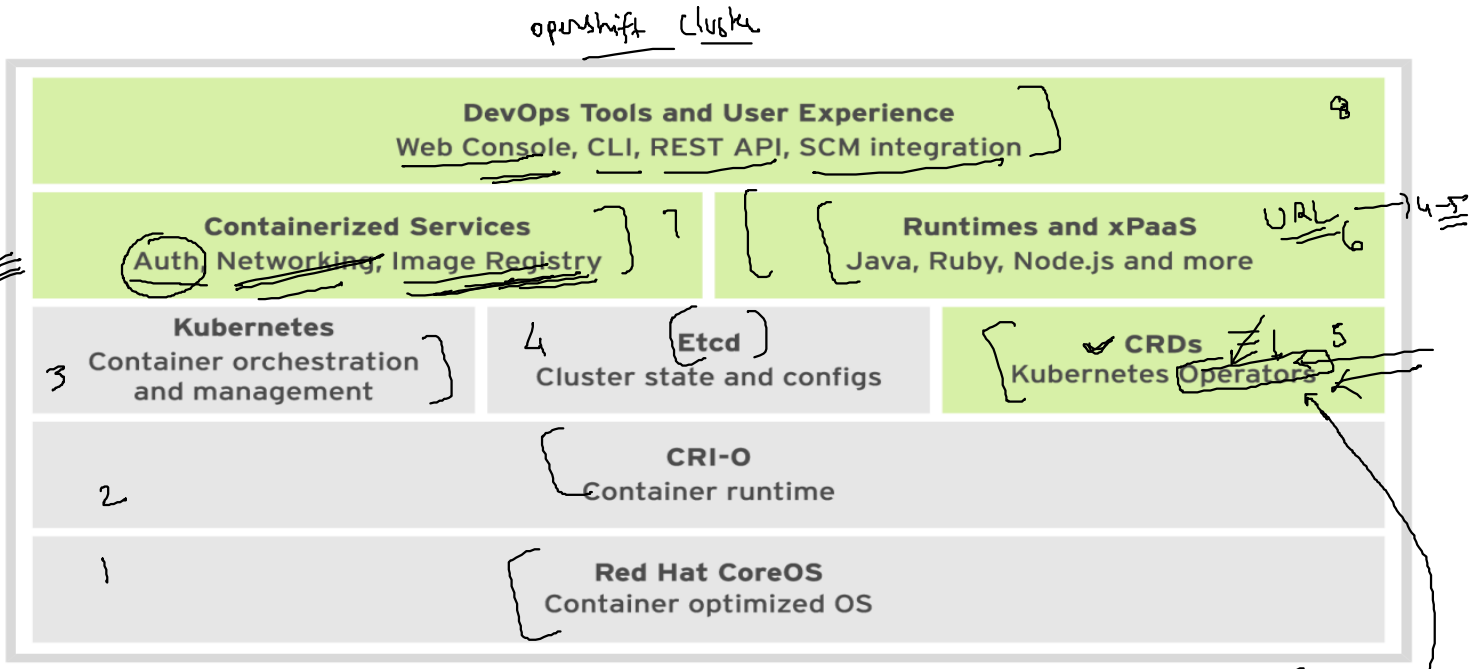
RAM, Storage, CPU

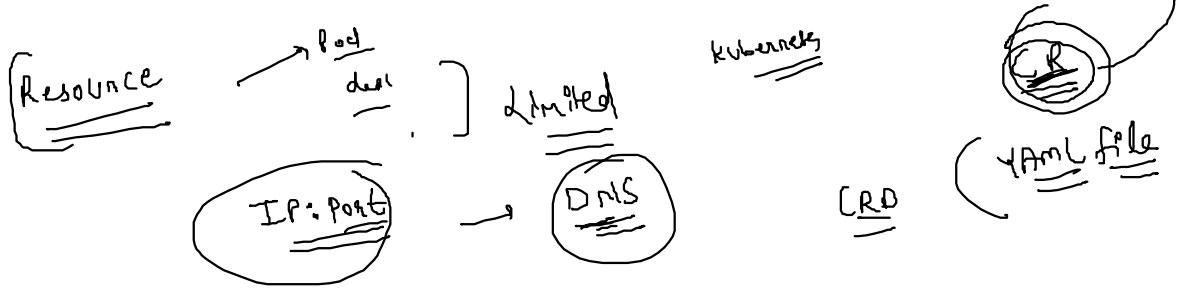
1. kube api server
2. etcd server
3. Kube scheduler

4.x → (7)

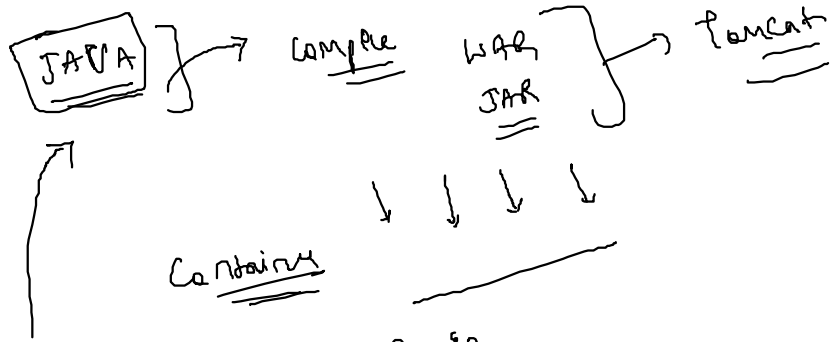
Openshift → 3.x → Security

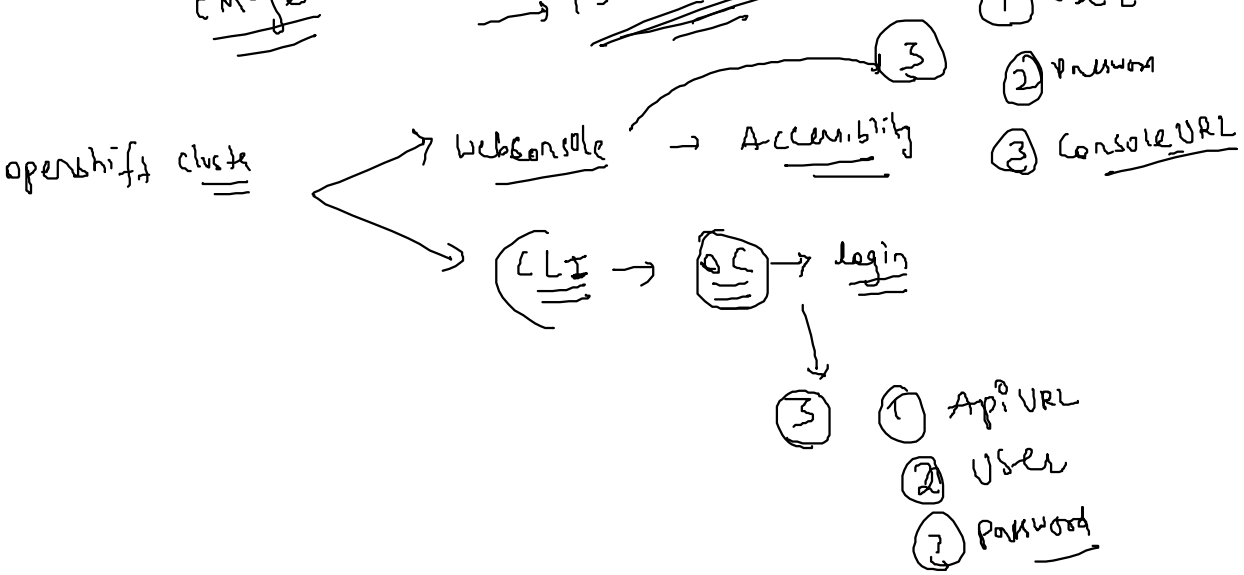
added some extra feature


openshift cluster
DevOps Tools and User Experience
Web Console, CLI, REST API, SCM integration
Containerized Services
Auth, Networking, Image Registry
Runtimes and xPaaS
Java, Ruby, Node.js and more
Kubernetes
Container orchestration and management
Etcd
Cluster state and configs
CRDs
Kubernetes Operators
CRI-O
Container runtime
Red Hat CoreOS
Container optimized OS

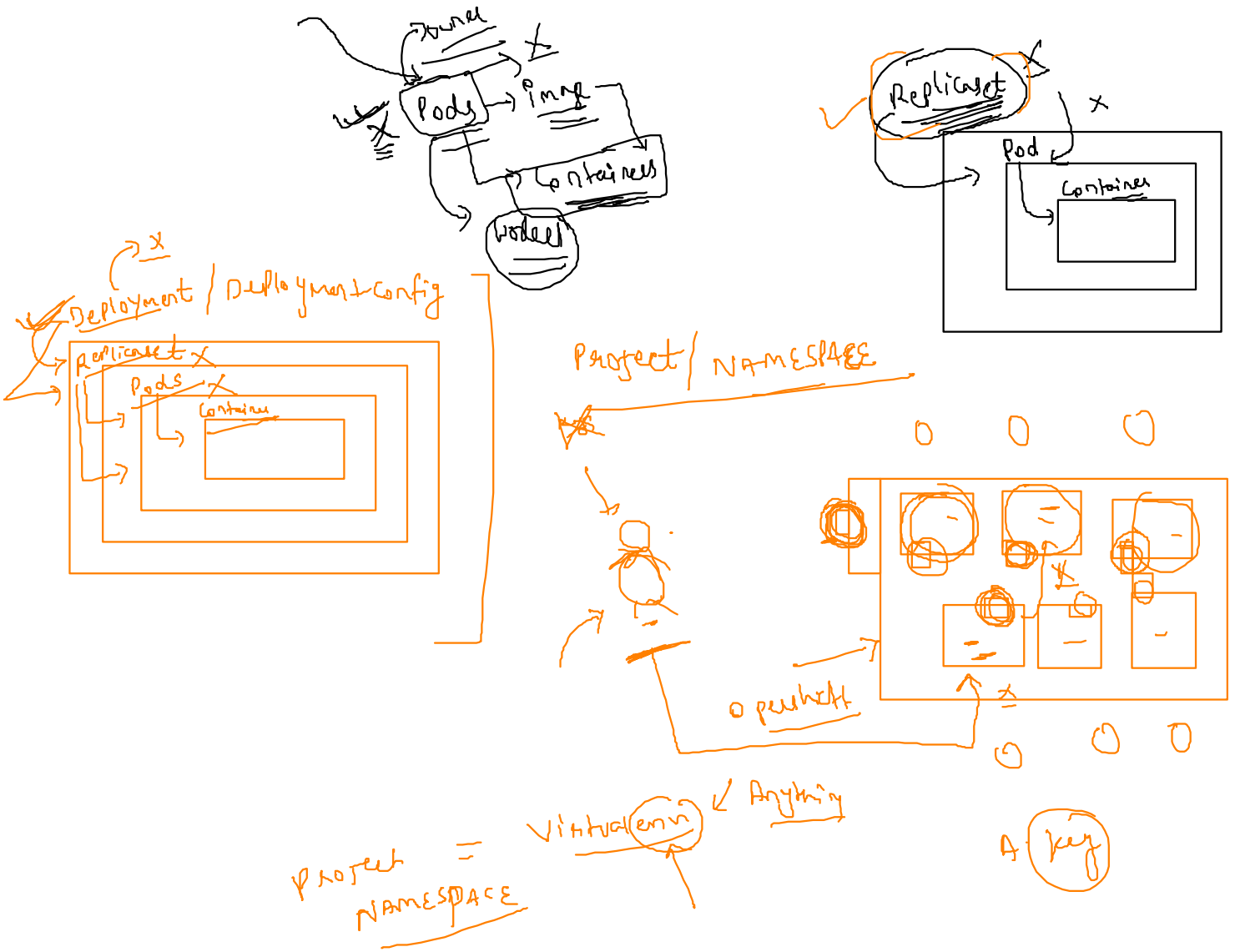

Resource
Pod
deal
Limited
IP:Port
DNS
CR
CRD
YAML file


dockerhub
Private Registry


JAVA
compile
WAR
JAR
Tomcat
Container
Image
15-20min

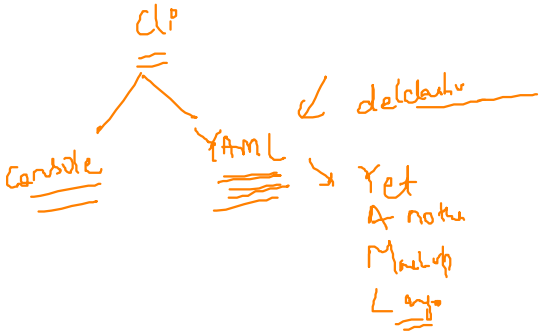

openshift cluster
Webconsole
Accessibility
CLI
oc
login
1 User
2 Password
3 Console URL
3
1 Api URL
2 User
3 Password

openshift Resources → Pods



Create project namespace ---- project name should be your login username user1 user3

loop --- givve



1. Deployment → image → Pod → Container → 1 App

2. Deploy → image:v2 → load — Conti → 2 App

1 App

front → 1 d

backend → 2 d

Services

Pod / deployments → CLI console

YAML = key:value

Pod - YAML = Generate it

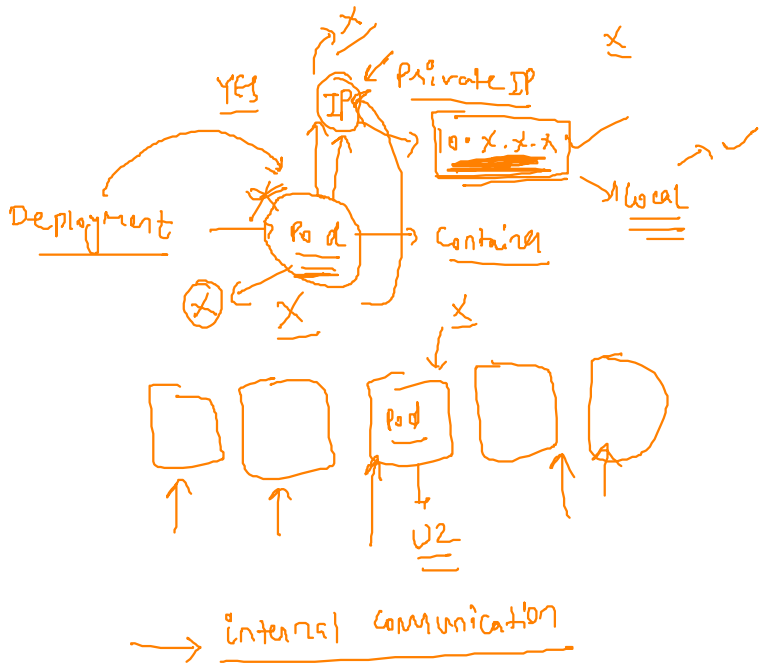
Services

11.12.13.14 pub → Public

130.131.132.233

~~141.241.255.247~~

~~171.172.272.273~~



① 10.X.X.X private

② 192.16.X.X — 172.31.XX

③ 192.168.X.X

→ internal communication

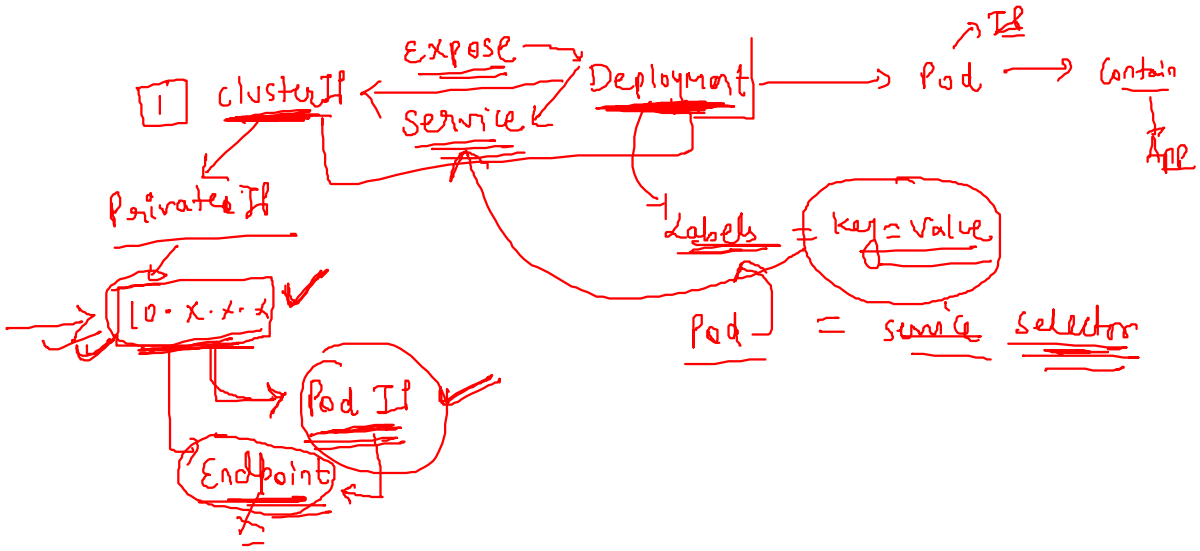
Resource

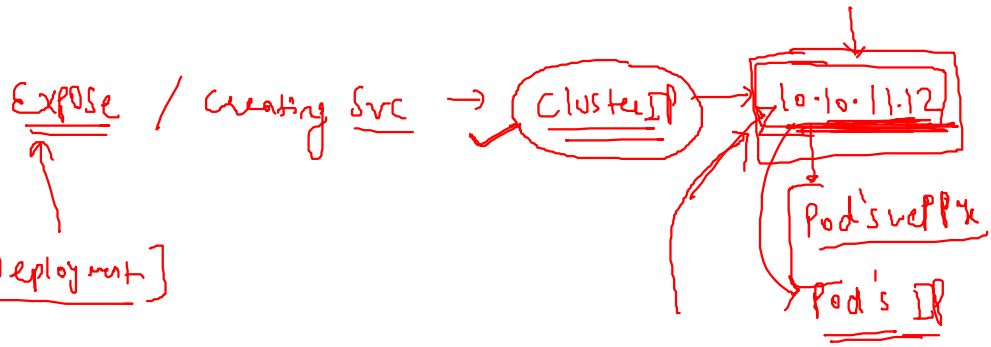
Services → Expose App

4 types

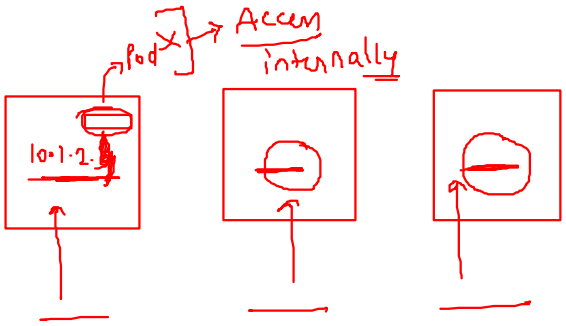
1. Cluster IP ✓
2. Nodeport ✓
3. LB ✓
4. External NAME ✓

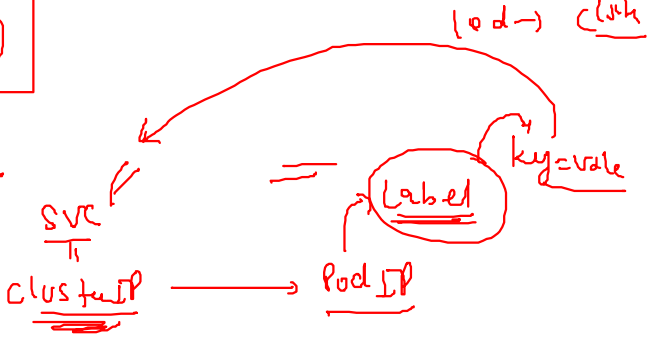
1 Relies on internal communication





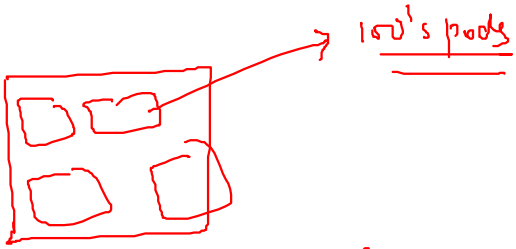
1-de — 1-Cls

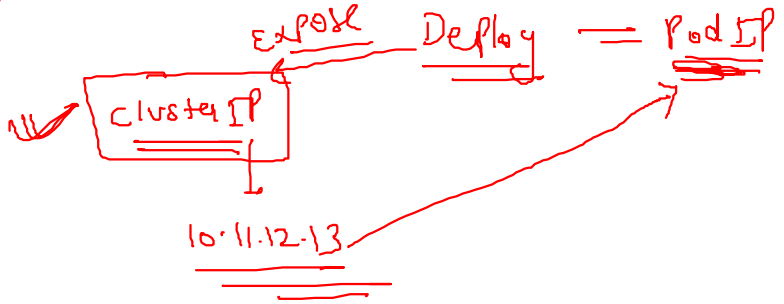




3dep → Expos

3-clu2





expose — D2 → Pod IP

Cluster IP — 10.12.11.14

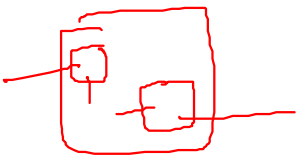
unique

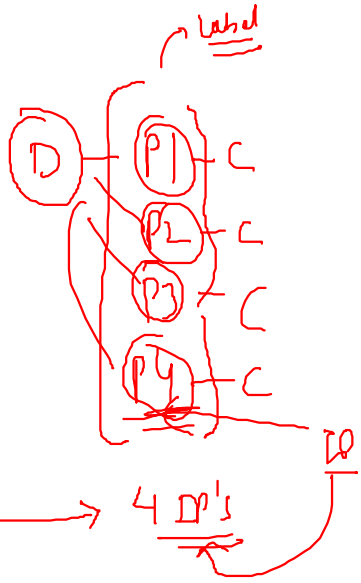
— Pod1 = a=b

— Pod2 b=c

— Pod3 a=b

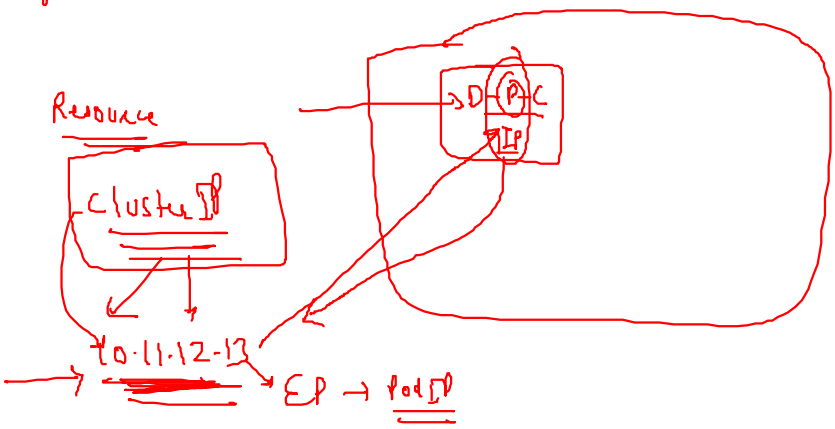
Micro → 6 label





EP

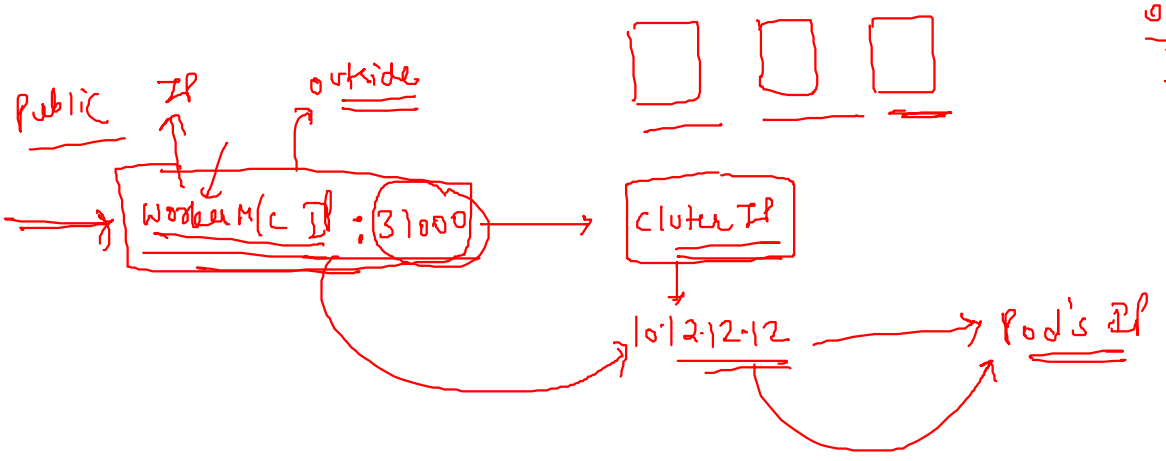
10.12.13.14



2 — NodePort → 30000 - 32767

M/c

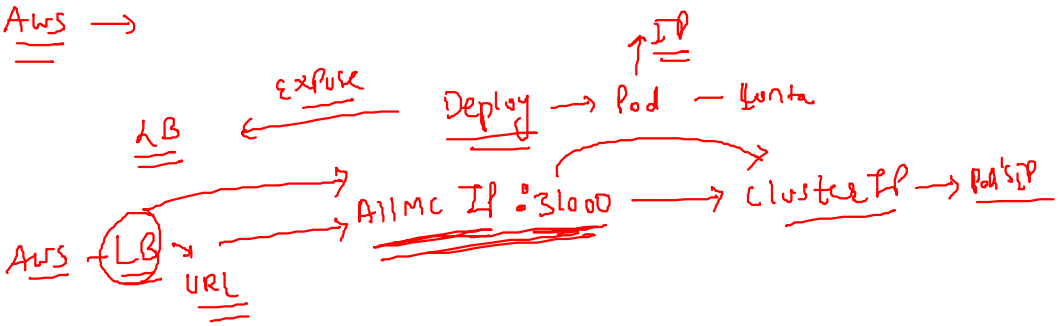
Port no..



open port

31000

3 Load Balancer → AWS →

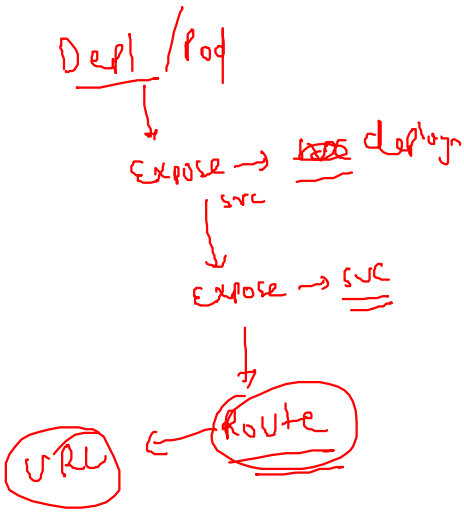


4 [External Name]

CLI / YAML / Console

[ClusterIP]
[NP]
[LB]

Resource



SVC selector ===== Pods label