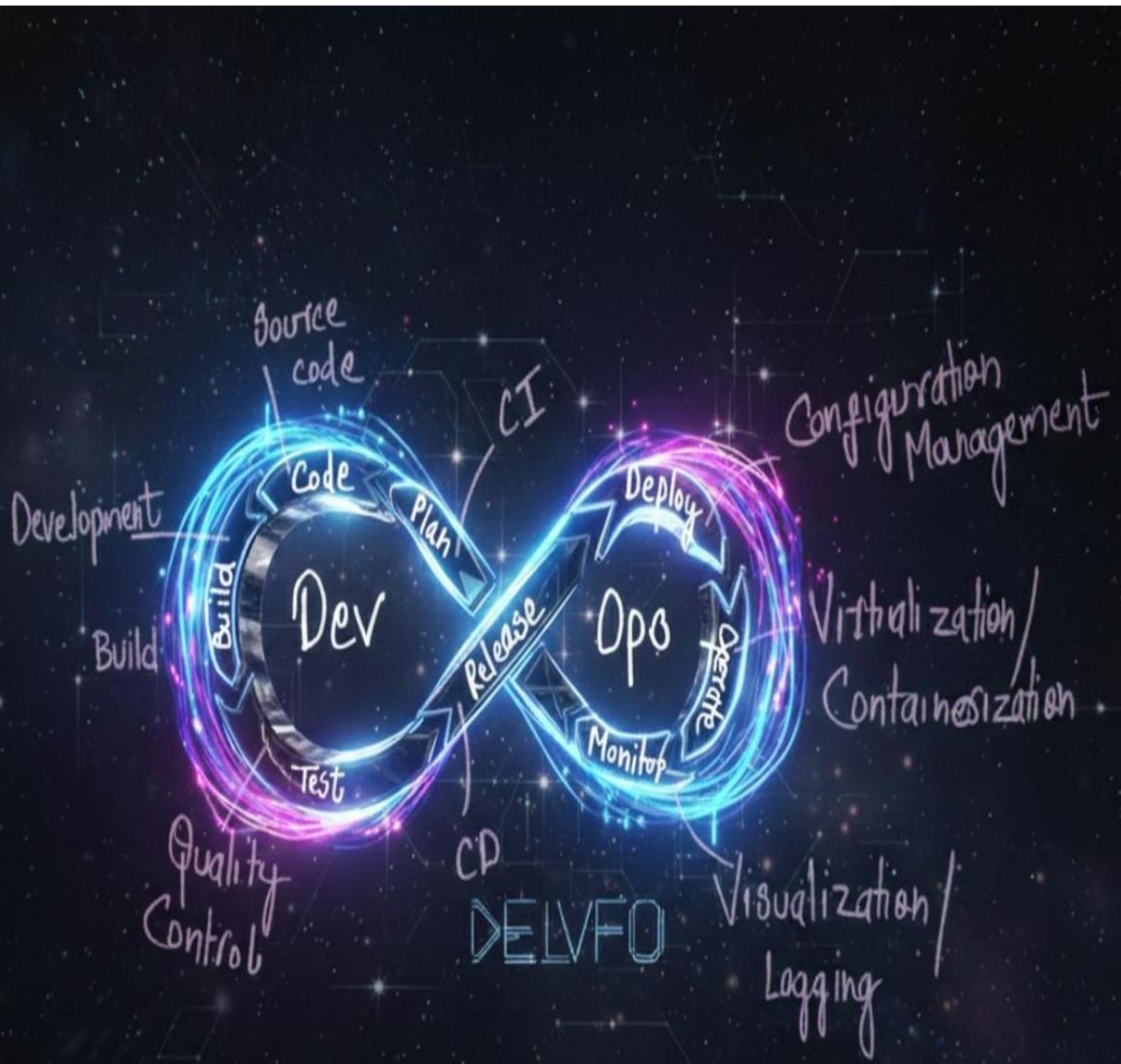


# DEVOPS

# NOTES



# Day - 1

## Fundamental of DevOps

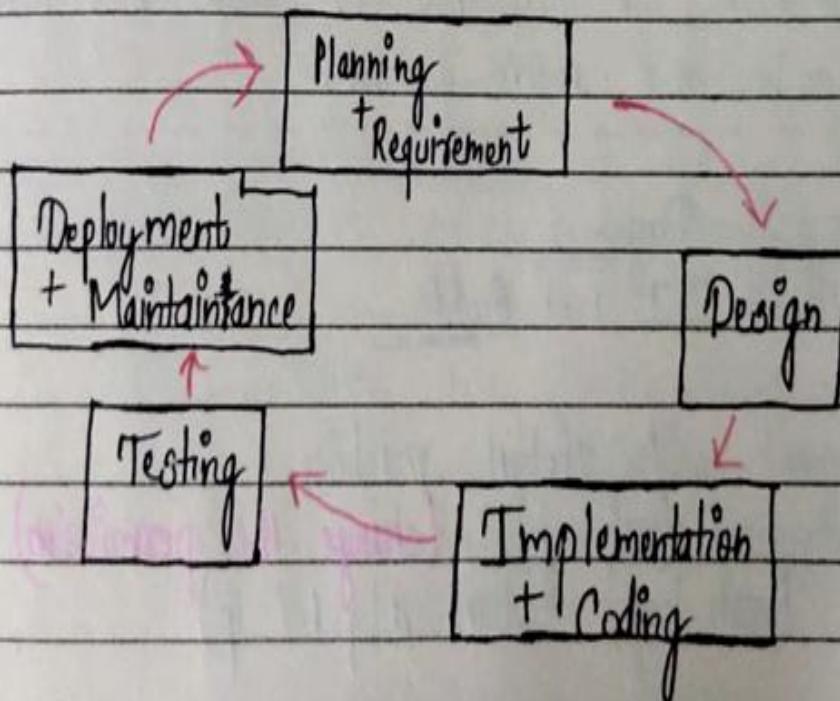
# What is DevOps ?

- ↳ IT is a culture that improves delivery of applications / faster.
  - ↳ • Automation
  - ↳ • Quality
  - ↳ • Continuous monitoring
  - ↳ • Continuous Testing

# Day - 2

## Improve SDLC with DevOps

SDLC (Software Development Lifecycle) is a practice in an IT industries to Design, Develop and Test the high quality software products.



## Day - 0 Virtual Machines

A virtual machine (VM) is a digital version of physical computer. It works same as physical computers and generally used in software industries.

### # Hypervisor :-

A hypervisor is a software that you can use to run multiple virtual machines on a single physical machines. (Resource sharing or divide)

## Day - 4 AWS or Azure - How to Create Virtual Machines

In this lecture, we learn how to create a virtual machines on aws and azure portal.

## Day - 5 AWS CLI Full Guide

### # How to connect with virtual machine :-

- ↳ chmod 600 (ash key). {change the permission}
- ↳ ssh -i (ash key) ubuntu@ (public ip)

## Day 7 Linux & Shell Scripting

# Linux is better operating system than windows because -

- ↳ FFIO free
- ↳ Open source (source code available)
- ↳ Secured.
- ↳ Multiple - Distributions
- ↳ Faster.

System, Software

Shell  
Kernel

### # Some shell commands:-

- ↳ ls - list      ↳ pwd → present working direct.      ↳ Hardwre
- ↳ cd      ↳ ls - ltr      ↳ touch → to create file
- ↳ nano or vi → are editors!
- ↳ cat → to show the file content.
- ↳ mkdir → create directory
- ↳ rm → delete files      ↳ rm -r → remove directory
- ↳ nproc → no. of process      ↳ df -h → disk size.

Day 7

### Live AWS Project using Shell scripting

In this lecture we learn about the aws project using shell scripting to report the usage of aws in a project.

Day 8

### Shell Scripting Project Used in Real Time Github API Integration

In this lecture, we learn about how grant/revoke access and list of people access github repository using

API integration. The shell script has been created to communicate with Github API.

### Day → 9 Git and Github

The version control system is used to track the data of source code which is easily manageable by developers.

### Version Control System

#### Centralised

- ↳ SVN

- ↳ CVS

(All the data present  
on a central server)

#### Distributed

- ↳ Git

(The copy data or  
data is distributed)

In the .git folder :-

- ↳ objects → all files are tracked as object.

- ↳ hooks → all the run automatically every time an event occurs in git repository.

- ↳ config → all configuration and credentials are stored.

- ↳ HEAD → returns the latest commit.

git diff → Check the changes of files

git log → Check all the commits.

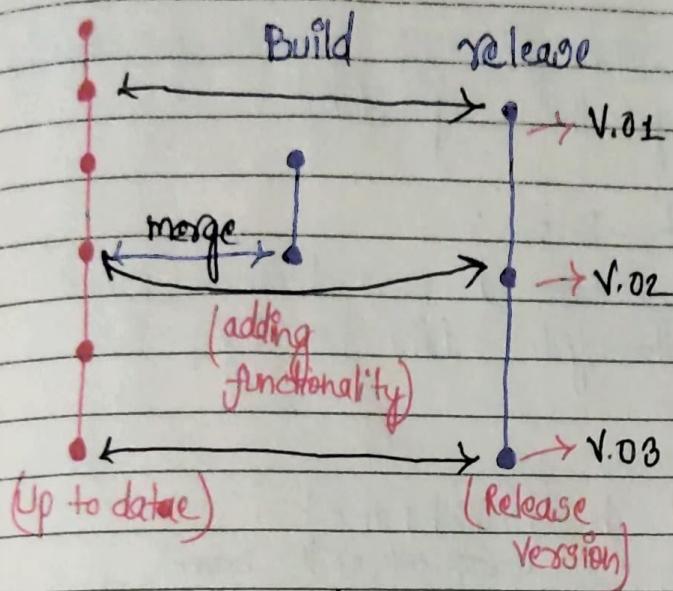
Day → 10

## Git Branching Strategy

# Branching :-

↳ Separation of functionality of new code.

master/main



The main/master branch is up to date when new functionality added or at release time. The build branch is used when you have to add new functionality. When the application deployed we use release branch.

Day → 11

## Git Interview Q & A and commands for DevOps

In this lecture we learn about basic commands of git for devops. (git add, git commit, git push)

# Git clone using SSH :- To cloning the git repository using ssh we need public key

↳ ssh-keygen -t rsa

(If creates a .ssh folder copy the public key in github / settings / SSH / New SSH key.)

# Clone vs Fork:-

↳ Clone is download the files of repository. when fork is used to cloning the repository.

# Git branch vs checkout - b :-

↳ git branch just create a new branch where git checkout - b gets the file data (existing) to create branch.

git (main): git checkout - b (branch-name)  
(source code of this branch).

# Git merge vs Git rebase vs Git cherry-pick

git (main): git cherry-pick (commit-id)

git (main): git merge feature branch (branch-name)

git (main): git rebase (branch name)

cherry-pick commit all the branches to main branch directly without any confirmation.

git merge merge all the branches and create a new commit (a cycle order)

git rebase merge all the branch and track the history of all commits. (Linear order)

Day 12

## Deploy and expose your First App to AWS

In this lecture we deploy an application on azure / aws platform.

Note :- Create IAM (Identity Access Management) roles before deploying a project.

#Important :-

- (i) If we deploy node.js application the procedure is same as local machine.
- (ii) Create a inbound rule in a security section to expose the port publicaly.

Day 13

## Top 15 AWS Services for DevOps

- (i) Amazon EC2 (Elastic Computing) (virtual machine)
- (ii) VPC (Virtual Private Cloud) (Security)
- (iii) EBS (Elastic Block Store) (volume or storage)
- (iv) S3 Buckets (Simple Storage Bucket) (storage)
- (v) IAM (Identity and Access Management) (permission)
- (vi) Cloud Watch (Monitoring)
- (vii) AWS LAMDA (Serverless cloud)
- (viii) Cloud Build Services (Manage CI and provide package of code).
- (ix) AWS Configuration (record of configuration EC2).
- (x) Billing & Costing (cost management)
- (xi) AWS KMS (Key Management Service) (cryptography keys)
- (xii) Cloud Trail (recording actions taken on account).
- (xiii) AWS EKS (Elastic Kubernetes Service) (automate deploy and manage)

- (xiv) Fargate, ECS (Elastic Container Service)  
 (xv) ELK (ElasticSearch, Logstash and Kibana)  
 (Analytics & search functionality)

## Day → 14 Configuration Management with Anible

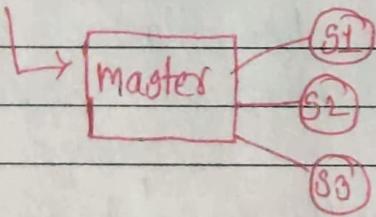
To manage the multiple servers at a same time it requires the configuration management.

(Ansible is the most popular for configuration management tool).

### Puppet

- \* Pull method
- \* Master Slave

Architecture

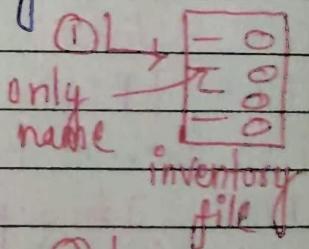


- \* Puppet language

### Ansible

- \* Push Method

- \* Agentless



- \* Support with Linux and Windows

- \* Simple (yaml project)

# Ansible uses ? → Python language  
# Platform ? → Support both Windows & Linux  
# puppet vs Ansible ↴ or Why it is used?  
(Windows) { } (Linux)

~~Day +15~~  
Ansible Zero to Hero

# Installation :-

- ↳ sudo apt install ansible
- ↳ ansible --version

# Generate public / private key :-

↳ ssh-keygen

# Ansible ad hoc commands vs ansible playbooks :-

Ansible ad hoc commands are one-line Linux shell commands and playbooks are like a shell-script, a collective of many commands with logic.

(★ Verbose → Verbose is used to show the list of operations perform on backend.  
(Similar command like debugging) -v).

# Ansible Roles :-

↳ Ansible roles provides a well-defined framework and structure for setting your tasks, variables, handlers, metadata, template and other files.  
- ansible-galaxy role init (name).

Day → 16

## Infrastructure as Code

Infrastructure as Code (IaC) is the managing and provisioning of infrastructure through code instead of through manual processes.

(The problem arises in a IaC that it will require to learn different tool with different cloud provider.)

(That's why HashiCorp created a terraform which enable you to manage infrastructure with minimal changes.)

# Terraform uses API as a Code :-

↳ API key is a code to identify and authenticate an application of user.

(Terraform convert the script into API for calling different cloud providers.)

Day → 17

## Everything about Terraform

Terraform is IaC (Infrastructure as Code) software tool created by HashiCorp which is used to automate various infrastructure tasks.

## # Main Commands :-

- (i) init → prepares your working directory.
- (ii) validate → check whether the configuration is valid.
- (iii) plan → Show changes required by current configuration.
- (iv) apply → create or update infrastructure
- (v) destroy → destroy previously created infrastructure.

## # How to run terraform file to create a infrastructure as Code :-

- Connect to the cloud provider (azure, aws etc..)
- Run the terraform file with commands  
(Configuration should be proper else errors.)

(\*Note :- The best practice of terraform will created with multiple → main.tf, input.tf, variable.tf)

(\*Note :- The terraform state file should not be stored in repository and local system).

(\*Note :- Store the terraform in a remote backend  
ex → cloud provider storage)

## # There are many problems in terraform:-

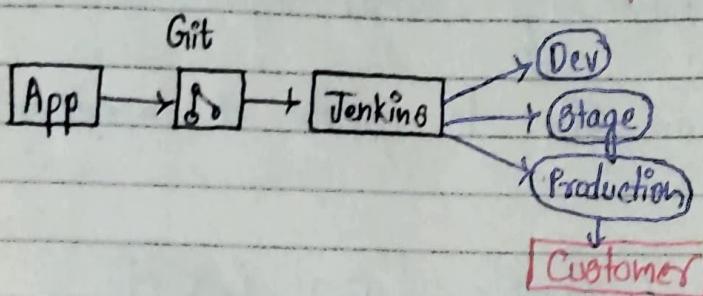
- ① It is very complex and difficult to manage.
- ② Manual changes cannot be identified and corrected.

Day → 18  
What is CI/CD?

CI/CD stands for continuous integration and continuous delivery / deployment.

It's a set of principle which automate software development process to deliver software change to users in a timely, repeatable and secure manner.

(\* Jenkins is used in CI/CD pipeline orchestrator).



(\* Github actions is a modern day CI/CD pipeline)

Day → 19  
Jenkins Zero to Hero

# Jenkins is based on Java, that's why JDK is required to run Jenkins

→ sudo apt install openjdk-11-jre

# Install Jenkins from the official website through cli or setup.

# If you are running Jenkins on virtual machine you have to set inbound rules to access

the software. To check:-

→ ps -ef | grep jenkins

\* Default port of jenkins is 8080)

# For getting the administration password copy the path and open in terminal from getting started page

→ sudo cat (path)

\* Working Docker as an agent:-

# Installation :-

→ sudo apt install docker.io

# If you install jenkins in a admin then you need to grant permission to user to use docker:-

→ sudo su - usermod -aG docker jenkins usermod -aG docker ubuntu systemctl restart docker

# Install docker pipeline in a jenkins plugin.

# Once you done with setup, restart jenkins :-

{ → http://<ec2-instance-public-ip> // restart

Jenkins is an open-source automation server which enables developers to build, test and deploy their software.

### Day $\Rightarrow$ GitHub Actions | Actions vs Jenkins

Github actions is a CI/CD tool that allows you to automate your build, test and deployment pipeline.

#### #Advantages :-

- **Hosting**  $\rightarrow$  Jenkins is self hosted it require servers, Github action hosted by github and run directly on repository.
- **User Interface**  $\rightarrow$  Jenkins is complex UI , Github action has streamline and user-friendly.
- **Cost**  $\rightarrow$  Jenkins can expensive to run and maintain it require resources, where Github Actions is free for public repository and tiered pricing model for private repository.

#### #Disadvantages :-

- **Integration**  $\rightarrow$  Jenkins can integrate with wide range of tools and services, while Github Actions is tightly integrated with Github platform.

## Day → 21 CI/CD Interview Questions

In this lecture there are some CI/CD interview questions..

(All questions & answers are in Git repository)

- Q.1) Can you explain CI/CD process in current project?
- Q.2) What are different ways to trigger Jenkins?
- Q.3) How to backup Jenkins?

## Day → 22 Project Management tools for DevOps

# Agile Methodology is generally used in project management:-

Agile methodology is a project management approach that involves breaking the project into phases and emphasizes continuous collaboration and improvement.

(# JIRA Software is an project management tool that supports any agile methodology, be it Scrum, Kanban or your own unique flavours).

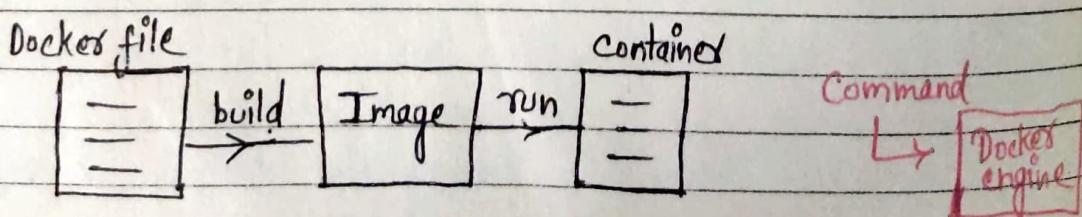
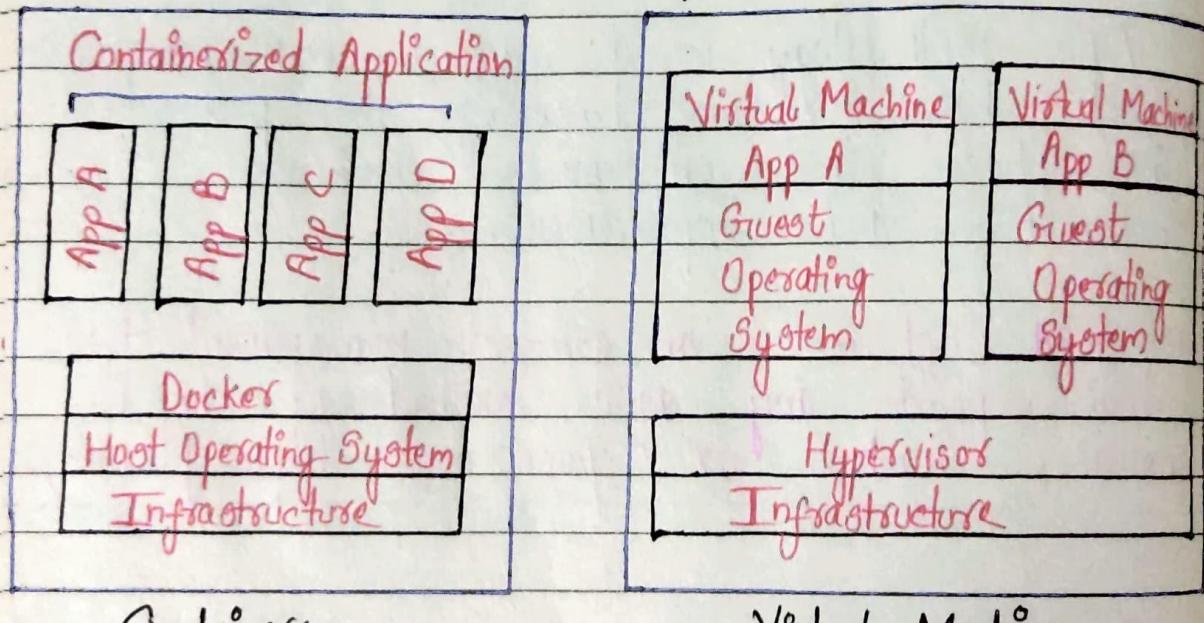
## Day → 23

# Introduction To Containers

A container is a standard unit of software that packages up code and all its dependencies so the applications run quickly and reliably from one computing environment to another.

### # Docker Containers :-

A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.



Day + 24

## Docker: Zero to Hero

### # Docker :-

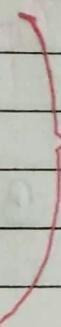
↳ A docker is a containerization platform that provides easy way to containerize your application, you can build, run, push the container to registries such as Dockerhub, Quay.io etc.

### # Docker important commands :-

- (i) docker build → builds docker images from dockerfile.
- (ii) docker run → run containers from docker images.
- (iii) docker push → push the container images to public / private registries.

### # Some terminology of Docker :-

- (i) Docker Daemon
- (ii) Docker client
- (iii) Docker Desktop
- (iv) Docker registries
- (v) Dockerfile
- (vi) Images



All the terminology all describe in github repository.

### # Installing Docker on VM :-

→ sudo apt install docker.io

→ sudo systemctl status docker

↳ check whether docker is running or not.

→ sudo usermod - aGs (username)  
↳ give permission to user (-rectest).

→ docker build -t (location) :latest  
↳ tag (Same as commit id)  
↳ same directory

→ docker run -it (location) :latest

(Use the correct tag format)

→ docker tag local-image:tagname new-repo:tagname

## Day → 2.5 Docker Containerization for Django

In this lecture, we learn about how to containerize a Django application.

(If you have a programming language knowledge, then it is same as previous video.)

## Day → 2.6 Multi Stage Docker Builds

# Multi-stage builds :-

There are two reasons to use multi-stage builds :-

- They allow you to run build steps in parallel, making your build pipeline faster and efficient.

- They allow you to create a final image with a smaller footprint, containing only what's needed to run your program.

## # Distroless images :-

Distroless images contain only your application and its runtime dependencies.

Day → 27

## Docker Volumes and Bind Mounts

volume

### # Docker ~~file~~ :-

A docker ~~file~~ volume is an independent file system entirely managed by Docker and exists as a normal file or directory on the host, where the data is persisted.

### # Docker bind mount :-

A bind mount, a file or directory on the host machine is mounted on a container.

### # Creating a volume :-

→ docker volume create (filename)

- docker volume ls

↳ List the volumes

- docker volume inspect (filename)

↳ Details of volume

# Docker - Delete Volume :-

- docker volume rm (filename)

# Creating docker mount :-

- docker run -d --mount source=(volume), target=/var/www/html nginx:latest

(\*: After creating a mount in a volume it's not get deleted because it's in use.)

Day → 28  
Docker Networking | Bridge vs Host vs Overlay

# Bridge Network :-

A bridge network uses a software bridge which lets containers connected to the same bridge network communicate, while providing isolation from containers that aren't connected to the bridge network.

# Host Network :-

Host networking is a networking mode in which a Docker container shares its network namespace with the host machine.

+ docker run -d --name (container) --network=host

## # Overlay Network :-

Connects multiple Docker daemons together to create a flat virtual network across hosts.

## # For login or executing the container :-

→ docker exec -it (container-name)

## # Ping Command for Linux :-

→ apt - get install iputils-ping -y

→ ping (ip address of another container)

↳ To check whether it is connected or not.

## # Docker network Commands :-

→ docker network ls → List all networks  
→ docker network rm (network name) → Remove network  
→ docker network create secure-network → create network  
(name)

→ docker run -d --name (container-name) --network=   
(network-name) nginx:latest → (package)

↳ Attach private network to container.

## Day → 29 Docker Interview Questions

In this lecture we learn about 12 important docker interview question which will help in the interviews.

## Day → 30 Introduction to Kubernetes

Kubernetes, also known as k8s, is an open-source system for automating deployment, scaling, and management of containerized application.

### # Why use of Kubernetes :-

- Service discovery and load balancing
- Storage orchestration
- Automated rollouts and rollbacks
- Self Healing

### # Kubernetes vs Docker :-

#### Kubernetes

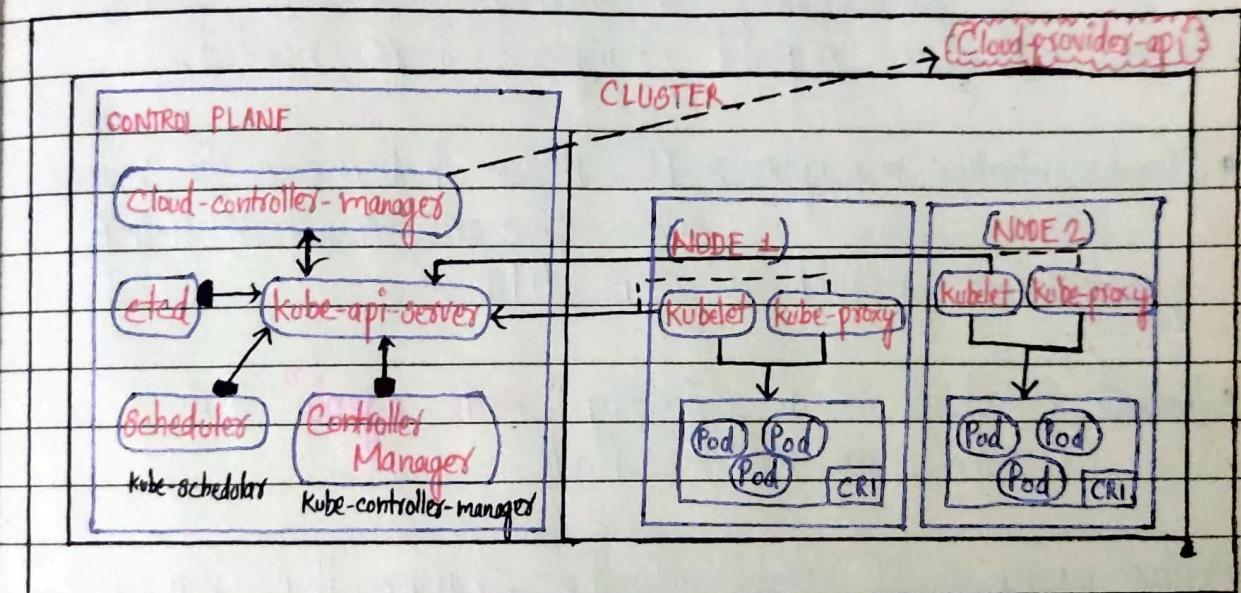
(i) Kubernetes is an open-source platform used for maintaining and deploying a group of containers.

#### Docker

(i) Docker is a tool that is used to automate the deployment of application in lightweight containers.

- (ii) It supports auto-scaling of the containers in a cluster.
- (ii) Docker doesn't support auto-scaling.
- (iii) Hard to setup & configure.
- (iii) Docker's setup & installation easy.
- (iv) In practice, Kubernetes is most commonly used alongside Docker for better control of containerized application.
- (iv) With Docker, multiple containers run on the same hardware much more efficiently than the VM environment.

## Day 3 Kubernetes Architecture



- Nodes + A node may be virtual or physical machine, depend on cluster. Each node is managed by control plane and contain the services to run pods.
- Control Plane + A control plane manages clusters and resources such as nodes and pods.

- API-server → API-server validates and configures data for the API objects which include pods, services, replication controllers etc.
- Scheduler → Scheduler is a control plane process which assigns pods to nodes.
- etcd → etcd is an open source distributed key-value store used to hold and manage the critical information that distributed system need to keep running.
- Controllers → Controllers provides all the critical functionality it manages cluster's state then make or request changes where needed.
- Cloud controller manager → It allows Kubernetes to manage some cluster resources using infrastructure provider's own APIs.
- kublet → kublet is a primary "node agent" that runs on each node.
- kube-proxy → It monitors the changes that happen to service objects and their endpoints.
- Container-runtime → It is responsible for managing the execution and lifecycle of containers within the Kubernetes environment.

Day → B2

## How to manage k8s Clusters

In this lecture, we learn about how to manage 100s of Kubernetes clusters and their problems.

# kOps :-

↳ kOps is an open-source project which helps to create, destroy, upgrade and maintain a highly available, production-grade Kubernetes cluster.

# kubeadm :-

↳ kubeadm is a tool built to provide kubeadm init and kubeadm join as best-practice "fast-paths" for creating Kubernetes clusters.

(# Note :- If you are working on cloud and use external tools ex → Kubernetes not EKS (amazon) then if you got an error then you won't get any help from amazon because you don't pay for it).

Day → B3  
Kubernetes Pods

A pod is a group of one or more containers, with shared storage and network resources, and a specification for how to run the containers.

# kubectl :-

↳ kubectl allows you to run commands against Kubernetes clusters.

## # Installation of kubectl :-

↳ Different commands for Linux, Windows or Mac  
is present on Kubernetes Website for installation.  
~~kubectl version~~

We are working on a minikube to create a pod  
→ winget install minikube

## # Start the minikube cluster :-

→ minikube start

If you want to start minikube as docker  
driver default then :-

→ minikube start --driver=docker

→ minikube config set driver docker  
default

→ kubectl get node (list all nodes)

## # Create a yaml file then :-

→ kubectl create -f (filename)

→ kubectl get pods (-o wide) List of pods.

→ kubectl delete pods nginx (filename)

## Day → 34

### Kubernetes Deployment

A Kubernetes Deployment tells Kubernetes how to create or modify instances of the pods that hold a containerized application.

→ `kubectl apply -f (deployfile)`  
 ↳ After creating a file

→ `kubectl get (deploy) (rs) (pod)`  
 → `kubectl delete (pod file)`

↳ If will delete and automatically create new pod.  
 (Auto Healing)

## Day → 35

### Kubernetes Services

In a autohealing concept, the Kubernetes immediately created a new pod after the previous go down. But the problem is it is create a new pod with new ip-address which has to accessible by user is difficult.

# Kubernetes Services :-

Kubernetes services is a group a set of pod endpoints into single resource?

- ↳ ① Load Balancing
- ↳ ② Service Discovery
- ↳ ③ Expose to world.

## # Load Balancing :-

↳ Load Balancing tracks the availability of pods with the Kubernetes Endpoints API.

## # Service Discovery :-

↳ Service Discovery is the process of connecting pods and services.

↳ Labels & Selectors → A group of primitive that allow logic operation on object in Kubernetes.

## # Expose to world :-

- Cluster IP (default) → Inside the system.
- NodePort → Inside the organization
- LoadBalancers → External World.

## Day → Q6 Kubernetes Interview Question PART-1

In these lecture we learn about 10 kubernetes interview question which will help in interviews.

## Day → Q7 Kubernetes Services (Deep Dive)

While creating yaml file change label and selector name which is used in to create service.  
(Change the image and port in spec section according to Dockerfile content.)

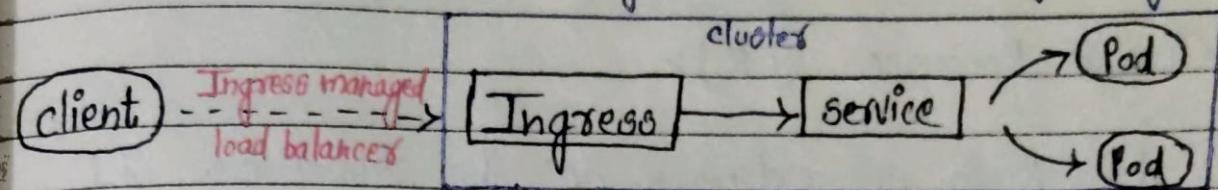
Create a service.yaml file change selector, targetport according to yaml file.

## Day → 38

### Kubernetes Ingress

#### # Kubernetes Ingress :-

Ingress exposes HTTP and HTTPS routes from outside the cluster to service within the cluster. Traffic routing is controlled by ingress.



#### # Why need of Kubernetes cluster ?

- Enterprise → security  
→ Load Balancing capabilities
- Services → IP (VM charges for IP addresses)

(Note :- you need ingress controller to use ingress services and need to update the host on /etc/hosts file)

## Day → 39

### Introduction to k8s RBAC

#### # k8s RBAC (Role-based Access Control) :-

RBAC is a method of regulating access to computer or network resources based on the roles of individual users within your organization.

- ↳ Service Accounts / Users
- ↳ Roles / Cluster Role
- ↳ Role binding / CRB

# Day +40

## Kubernetes Custom Resources

### # Custom Resource Group (CRD's) :- (Definition)

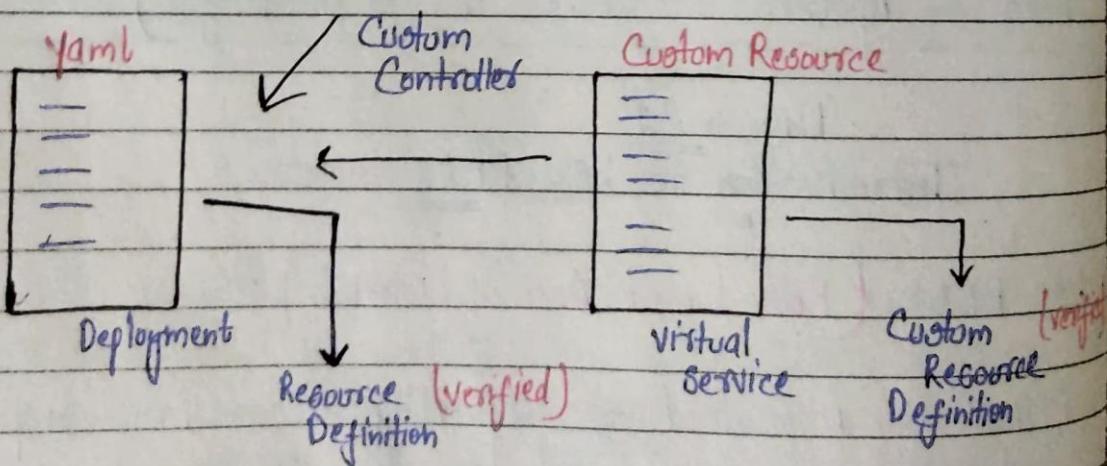
A CRD object creates a new object (custom resource) with name and schema. It extends the base Kubernetes API with any object.

### # Custom Resources (CR) :-

A CR is an extension of the Kubernetes API that allows you to store your own API objects and lets the API server handle the lifecycle of a

### # Custom Controllers :-

The Custom Controller keeps the current state of Kubernetes objects in sync with your declared desired state.



Day +41

## Kubernetes Live Project || Configmaps & Secrets

# ConfigMaps  $\Rightarrow$  A ConfigMap is an API object used to store non-confidential data in key-value pairs.

Pods can consume ConfigMaps as environmental variables, CLI arguments, or as configuration files as volumes.

# Secrets  $\Rightarrow$  A secret is an object that contains a small amount of sensitive data such as a password, a token, or key.

echo \$(encrypt-name) | base64 --decode

Day +42

## Kubernetes Monitoring Using Prometheus & Grafana

Kubernetes monitoring is a form of reporting that helps with proactive management of clusters. It ease the management of containerized infrastructure by tracking the utilization of CPU, memory and storage.

# Prometheus :-

$\hookrightarrow$  Prometheus is an open-source technology designed to provide monitoring and alerting functionality for cloud native environment, including Kubernetes.

## # Grafana :-

→ Grafana is renowned open-source analytics and interactive visualization platform that supports data presentation methods using pluggable panel architecture.

## Day → 43 AWS Live Project | Deploy App using HTTPD

In this lecture we learn how to deploy a web application on ~~aws~~ aws on apache server..

## Day → 44 The End of Free DevOps Course

Farewell and End of the course declaration.  
(Some Future Plans)

DevOps Course

Finished