**DevOps 🡪** It is a process of improving the application delivery by ensuring proper automation, code quality, continuous monitoring & continuous testing.

**Before DevOps 🡪** Developer 🡪 Central location 🡪 Server 🡪 Testing 🡪 Build and Release Engineer 🡪 Pre-prod 🡪 Production

**Introduce 🡪** I have 2.8 years of experience in this field, from last 8 months I’m working as a CloudOps admin & before that I have worked as a DevOps engineer for 2 years.

SDLC 🡪 Software Development Life Cycle

Design

Develop

Test

Planning(gather requirements) 🡪 Defining(documents-software requirement specification document) 🡪 Designing(High level design-high available, scalable, db, Low level design-modules, functions) 🡪 Building(automation) 🡪 Testing(automation) 🡪 Deploy(automation)

Building 🡪 Developing

Developer(DEV) writes code n later its reviewed by team members(peer) & later it will be uploaded on git

Testing(QA) application stored in git that is taken n deployed on server later Quality Assurance team do testing

Deployment promote to Prod n customer will receive in prod

DevOps fastens the process, improves the process to deliver the code quickly by ensuring above three things matched -> automated

Agile 🡪it won’t wait for full documentation, it will work in smaller parts

On one server we were able to deploy one application if server is having 100 GB Ram n 100 CPUs n application requires 4 GB RAM n 4 CPUs n remaining memory n cpu wasted it cause inefficiency, so to avoid this hypervisor came into picture.

So on top of physical server we can install hypervisor software using which we can create virtual machine which is logically isolated from physical server. 🡪 efficiency

For automation we can use AWS CFT(Cloud Formation Templates),AWS API, AWS CLI, AWS CDK(Cloud Development Kit), Terraform

Using AWS CLI you can interact with AWS n create any resource

Automated way

AWS CLI download

aws version

Which aws

/usr/local/bin/aws – it’s just binary it doesn’t have user information

So open console

User dropdown 🡪 security credentials 🡪 create access keys

Access key n secret access key copy somewhere

aws configure n enter details

open s3

aws s3 ls

aws s3 mb sumati

**OS** acts as a mediator between software n hardware

OS responsible for starting communication between software n hardware

Linux 🡪 free, opensource, secure, fast

OS 🡪kernel(heart of os) 🡪 system libraries 🡪 systemsoftware, userprocess, compilers

**Kernel’s** responsibility to establish communication between software n hardware

Kernel does :-

Device management

Memory management

Process management

Handling system

System Library:- Responsible for performing tasks

**Shell scripting**

Shell 🡪 the way you talk with operating system

bash

ssh –I pem \_filepath user@ip

ls

pwd(present working directory)

ls –ltr

touch sumati

vi test

cat test

remove dir 🡪 rm –r

remove file 🡪 rm

memory 🡪 free –g

CPUs 🡪 nproc

disk size 🡪 df –h

memory,cpu,disk 🡪 top

Automation is a process where you try to reduce manual activities

Shell scripting in linux is process of automating day to day activites

touch first.sh

Suffix **man** command with any command it will give details

vim/vi

shebang 🡪 #!

vi first.sh

#!/bin/bash

Bash, sh, ksh, dash these are the different executables which are used for diff machines

Executables which are responsible for executing shell scripts

each of them have their own syntax diff not much diff

What is diff between /bin/sh & /bin/bash

Previously both are same because /bin/sh was redirecting using linking concept to /bin/bash but now it is not the same because some of the

operating systems decided to use dash as default, so your script

might not execute if you are writing in bash scripting on a machine

where dash is default.

vi first.sh

#!/bin/bash

echo “my name is sumati”

vi Second.sh

#!/bin/bash

echo “Hello”

sh first.sh or ./first.sh

chmod 🡪 grant permission

ch – change

which are permissions for root user, which group, whatyou’re your permission

chomd 777 🡪 all permissions

man chmod

421 🡪 read write execute

user group everyone 🡪 chmod 444 🡪 all having read permission

history

How to create folders

mkdir folder

write comments using # in shell script

vi first.sh

#create folder

mkdir sumati

#create file

cd sumati

touch firstfile second file

chmod 777

always write metadata information(­­­­author, date, version)

debug mode(it will show which command it’s executing n showing output) 🡪 set –x

ps – processes, -ef – details about process 🡪 ps –ef

grep “amazon” – searching amazon, | - send output of first command to second command 🡪 ps –ef | grep “amazon”

vi second.sh

./second.sh | grep 1

Interview question

What will be output & why : date | echo “date is”

output: date is

date is default shell command it sends output to stdin standard input & pipe commad only receives information from the output of date command but date command sending output to stdin so | pipe it wont work or it wont redirect output of first command to second command & pipe command only receives output when command not sending output to stdin

vi test

grep name test

grep name test | awk -F” ” ’{print $4}’

ps –ef | grep amazon | awk –F” ” ’{print $2}’

while using pipe in script use 🡪 set –e(exit the script when there is an error, drawback-when we use pipe then it won’t show error), set –o pipefail

curl command retrieve information from internet

curl logfileurl | grep error

POSTMAN – helps to create api request

Man curl

Curl –x GET api.foo.com

Download logfile 🡪 Wget logfileurl

What is the diff between curl n wget

Curl command fetch files from internet & wget command download file

To find file(/- means find everything) 🡪 find / -name pem.d

Used for trapping signals,

not executing signals like not working control c 🡪 trap

trap “echo don’t use ctrl+c” SIGINT

Signals 🡪 control C signal, kill pid

Crontab(it’s like alarm) 🡪 linux admin users can schedule tasks or commands to run periodically at specified intervals. These tasks can be automated to run daily, weekly, monthly, or at other custom time intervals.

Report🡪 nodehealth

Softlink 🡪 we can create softlink for file using ln –s, if original file is deleted or broken then symbolic link is broken

hard link 🡪 We can create hardlink using ln, here nothing like original file if we delete original file then also it will point to same data

Interview questions

1. List some of the commonly used shell commands?
2. Write a shell script to list all processes
3. Write a script to print only errors from a remote log
4. Write a shell script to print numbers divided by 3 & 5 and not 15
5. Write a script to print number of “s” in Mississippi
6. How will you debug the shell script
7. What is crontab in Linux?Can you provide an example of usage?
8. How to open a read-only file?
9. What is the difference between soft and hard link?
10. What is the difference between break and continue statements in a loop?
11. What are some disadvantages of shell scripting?
12. What are the different types of loops and when to use?
13. Is bash dynamic or statically typed and why?
14. Explain about a network troubleshooting utility?
15. How will you sort list of names in a file?
16. How will you manage logs of a system that generate huge log files everyday?

Jq used for json parcer

API – Application Interface

Getting information directly on web that is user Interface

Interacting with applications programmatically that is API

We can talk to application using API, CLI

Instead of logging in github we can create script n integrate it with github to give access to users

Git fork 🡪 creating copy of original repo with entire code means if original repo goes down then also we won’t face issues

Git Hub it is a web-based platform which is created on top of git which provides issues, usability, project management, reviewing, commenting

Centralised Version Control System

Cvs(concurrent version system)

Only one centralized location/server will be there n if it goes down dev can’t communicate n it requires network

Distributed Version Control System

GIT

We can create multiple copies of repos for which we call as fork n it can work offline also

git init

git add .

git commit –m

git log

git reset --hard commit id

git status

git-branch

Feature branch – if we want to add new feature like suppose uber is already there for cab n if you want to add bike as so you can create new feature branch n once code is completed n it’s correct then you can merge it with master branch n you can delete feature branch

Release branch – if you want to add some new code in short if you want to update version so you can create release branch n after completing release branch directly you can send it to customer for testing n on master branch other developers will be working actively sometimes you can merge release branch also with master

git push

git workflow 🡪 git add && git commit –m “msg” && git push

git remote –v

fork is used to create copy of repo n clone is used to download repo

git branch, git checkout –b

git merge, git rebase, git cherry-pick(we can check git log for division branch from master branch using git log branchname n we can add/merge code by using git cherry-pick commit-id, if there are less commits then cherry-pick is easy)

merge conflict: if many developers were trying to updating same file then it occurs, so you can remove conflicts from file n you can check with dev which changes should be there n after that save , add n commit file

git merge n git rebase will do same work n if you want linear commit history then go with rebase or else you can go with merge n merge will add commitids at top n rebase at it at bottom

AWS 🡪 it is a top cloud provider, it provides more than 200+ resources, it provides services like IAAS, PAAS

EC2 🡪 Elastic Cloud Compute 🡪 to create instances

VPC 🡪 security 🡪 we want to store data with privacy security so for that we can use VPC 🡪 various components are there subnets. Inbound outbound traffic

EBS 🡪 volume 🡪 snapshots n backup🡪 encrypt 🡪 with ec2 volume storage will come by default 🡪 but we were deploying some app or anything related to db then files were keep on creating so that time EBS volume is more useful you can detach them take some snapshots, backup n again reattach

S3 🡪 storage, encryption is needed 🡪 application read some data from excel or files that content that has stored somewhere so for that we can use s3

IAM 🡪 Identity access management 🡪 it will give access to user who all can read or write files

Cloudwatch 🡪 monitoring 🡪 all the actions which we perform on aws that we can monitor here 🡪 if dev missed to encrypt EBS storage then we can send them mail 🡪 when ec2 is crossing threshold amount of storage then using cloud watch we can send notifications

Lambda 🡪 serverless 🡪 it will directly run function or program it won’t ask for which machine you want n all it will directly create instance n execute function n after completing this it will tear down the instance 🡪see in above we seen dev missed to encrypt EBS so instead of creating ec2 we can directly run lamba function it will encrypt EBS n tear down instance.

Cloud Build service

AWS code pipeline 🡪 it works as Jenkins, means we have to write somecode like adding nodes n which action we have to perform on that node

AWS code Build 🡪 to complete code, run some tests n provide some software packages

AWS code Deploy 🡪 deploying applications on on premises or ec2 instances

AWS configuration service 🡪 suppose someone is creating unencrypted EBS n s3 bucket without versioning so we can can configure all this in AWS configuration n we can perform remedy action

Billing n costing

AWS KMS 🡪 Key Management Service 🡪 security 🡪 easy for you to create and control encryption keys used to encrypt your data.

Cloud Trail 🡪 enables governance, compliance, operational auditing, and risk auditing of your AWS account

AWS EKS 🡪 Elastic Kubernetes service 🡪 managed Kubernetes service provided by AWS that allows you to run Kubernetes clusters without needing to install, operate, or manage the Kubernetes control plane.

EKS simplifies the deployment, management, and scaling of containerized applications using Kubernetes on AWS infrastructure.

Fargate, Elastic Container service 🡪 AWS Fargate is a serverless compute engine for containers that allows you to run containers without managing servers or clusters.

Amazon Elastic Container Service (ECS) is a fully managed container orchestration service that allows you to run, stop, and manage Docker containers on a cluster of EC2 instances.

ECS can be used with Fargate as the underlying compute engine, providing a serverless experience for running containers.

Diff EKS vs ECS

EKS is a managed Kubernetes service, while ECS is a fully managed container orchestration service.

EKS allows you to run Kubernetes clusters, providing greater flexibility and compatibility with existing Kubernetes tools and workflows.

ECS offers a simpler and more integrated experience for running containerized applications on AWS, with native integrations with other AWS services like CloudWatch, IAM, and VPC.

EKS may be preferable if you have existing investments in Kubernetes or need advanced Kubernetes features, while ECS may be a better choice for simpler containerized workloads or if you prefer a more integrated AWS-native experience.

ELK 🡪 ELK is an acronym for Elasticsearch, Logstash, and Kibana, which are three open-source tools often used together for log management and analytics.

Elasticsearch is a distributed search and analytics engine used for indexing and searching large volumes of data in real-time.

Ansible 🡪 It’s a configuration management tool. It works on push mechanism. Means suppose you have to update some configurations on 10 servers then we can create playbook n you can push it to other servers, it’s agentless



Command 🡪 we can’t use pipe n redirection with command module

Shell 🡪 pipe will work here

Raw 🡪 run commands without python

API as a code is concept using which you can automate any provider

Infrastructure as a code 🡪 automate infrastructure 🡪 suppose if you want to deploy 300+ applications on AWS n n you written some 100+ CFT scripts to create resources like ec2, s3 n later you want to move it to azure n for azure again you need to write those script because here we were using Azure RM we can’t directly copy scripts from aws to azure so whenever you will try to migrate that time you will face this issue so to avoid this Terraform cam einto picture.

So suppose if you want to deploy apps to aws n you can write some terraform scripts n it will directly talk to aws api n it will make changes in script as per providers need, n suppose you want to migrate it to azure then also same same terraform script it wil make api call to azure n it will change n automate

CI 🡪 continuous integration 🡪 set of instructions executing before delivering app

CD 🡪 continuous delivery 🡪 set of instructions executing before deploying app

Unit testing --> to check functionalities working properly or not suppose you defined one function for addition of two numbers then it should to addition it won’t impact any other function

Static code analysis 🡪 analyzing code every functionalities syntax added properly n no extra variables added

Code quality/vulnerability 🡪 checking quality of code means if suppose yesterday only you lauched new version but today facing vulnerability that’s not good impact

Automation 🡪 above all are automation

Reports 🡪 how that functionalities working how I many tests are executed

Deployments 🡪 after successfully completing all above steps we can deploy app on some platform

Docker file 🡪 docker engine 🡪 image 🡪 container

Docker engine is single point of failure once it goes down we can’t create containers

Docker uses minimal operating system

On top of virtual machine we can create docker containerization platform n on top on that we can create containers

While using vms we are wasting so many resources not using ram cpus fully to avoid that docker containers came into picture

Docker is not secure compared to ec2

Containers are light weight in nature

They are having app + libraries + system dependencies

Containers use required resources from host operating system if some hacker he was accessing first container then he can able to access all containers because they were sharing same os

(bin/sbin/lib/usr/root/etc/var)Files part of container image those we can’t share with another container or else it will be like compromising security

Files used from os 🡪 networking stack,system calls, namespaces, control groups

client 🡪 docker host 🡪 registery

suppose if docker cli executes

docker run then docker daemon receives it it will create container

docker pull then docker daemon receives it will pull containers from registery

docker build then docker daemon receives it n it will create image

docker daemon is heart of docker

set of instructions(docker file run through cli) 🡪 after that docker daemon will receive it n it will build 🡪 image 🡪 docker run 🡪 it will create docker container

docker reduces complexity

sharing images with external world using docker hub(public registery)

registery is a platform to share images

docker wants permissions as root use

docker daemon runs only using root user

we can create n number of stages using multi stage docker build n it can be only having one final minimalistic image

it will reduce container image size 🡪 distroless images 🡪 security 🡪 not exposed to vulnerabilities 🡪 it will only have python runtime not even have find ls packages 🡪 not only we are reducing size but also ensuring containers executing securely

We can’t go back n check previous logs

If backend goes down then front end can’t read files

If by using cronjob we save host data on files which is present on host os that also container can’t access

To solve this docker comes with a concept

As a user Bind mounts 🡪 we can bind container dir with host dir

Volumes 🡪 offer better lifecycle 🡪 logical partition 🡪 create volume on host will be mounted to container 🡪 we can create so much of things with volumes that are using external resources

Networking allows containers to talk with each other n with host system

Both the containers having diff subnets they connect using bridge networking which is created by default while creating container 🡪 veth

Host networking 🡪 insecure way 🡪 when we creating container it will connect host with container

Overlay networking

Docker 🡪 on top of one single hosts 100s of containers will be running n container1 using more memory n it will kill 100th of container 🡪 application will goes down 🡪 container will come up automatically without users intervention that is auto healing but its not possible in docker 🡪 on top of single host we are running many containers definetly those containers affect each other 🡪 if some some occasion is there or launched some new movie then the people visiting Netflix site will be increased so we should have auto scaling feature load balancing so that it will equally balance load on all containers but docker doesn’t this features to solve all this four problems single host, auto healing,auto scaling, enterprise level support kubernetes comes into picture

Kubernetes 🡪cluster 🡪 group of nodes 🡪 suppose if due do some reason 99th container is facing issue going to terminate then kubernetes push it immediately to another node 🡪 replica set 🡪 if some festival is there n load will be increasing on site then it will automatically scale up containers n we have achieve that manually also by doing changes in yaml file 🡪 if some application is going down api server will get that notification n before the app goes down it will bring up new container 🡪 it provides enterprise level support

Kubernetes is container orchestration tool

Data Plane 🡪 executing actions 🡪 Kubernetes Worker node components 🡪 pod it’s like container 🡪 kubelet is responsible for making sure pod were running always it if it’s not running then some component from kubernetes control planewill be there it will automatically start 🡪 it have container runtime which runs container 🡪 kube-proxy it provides ip address, networking n default load balancing

Control plane 🡪 controlling actions 🡪 Kubernetes master node components 🡪 API server which acts as core compoents which takes incoming request n tells on which node it has to go, exposes kubernetes to external world 🡪 scheduler to schedule component on node we use scheduler , it is responsible for scheduling resources on kubernetes, whos decides info –api server, n who act’s on info that is kubelet 🡪 etcd backing store of entire cluster, key value store, objects 🡪 controller manager to detect n auto scale automatically replica set, the controller who ensures all the things are running that is CM 🡪 cloud controller manager if some user is requesting for storage from aws cloud providers then ccm it translates user request into api request that cloud provider understands

Kubernetes distributions are pre-packaged versions of the Kubernetes platform provided by various vendors and organizations. These distributions aim to simplify the deployment, management, and operation of Kubernetes clusters by offering additional features, tools, and support tailored to specific use cases and environments.

Kops is used to install kubernetes

Pods can be group of containers 🡪 so we can share storage networking files 🡪 containers talks using localhost when they are in same pod

Like for running docker we use some commands here in kubernetes we just have to update all commands in yaml file

Pods is wrapper for containers

Pod is like container only it won’t provide auto healing n auto scaling

Deploy it provides auto healing n auto scaling at the end after deployment it will create pod only

Replica set it ensures that in deploy yaml files the tasks give that are completed

Replica set controller who is actually working on auto healing n making sure if deployment says increase a pod by one that is done by replica set controller

the number of count of replicas depend upon the number of users trying to access your applications and also number of connections one particular pod can take

the number of count of replicas depend upon the number of users trying to access your applications and also number of connections one particular pod can take

before service 🡪 if suppose some app is there n three replicas created for that n it’s divided like 10 users can access one replica but what happens if some replica goes down but here it will bring automatically but the ip will change so user can’t access app

after service 🡪 load balancing 🡪 🡪 service discovery(labels n selectors) 🡪 n instead of accessing with ip address we can access app with labels means for every ip which is currently running going down coming up should come up with same label so if some one ip will go down then also won’t affect user

cluster ip 🡪 inside

nodeport 🡪 inside org

load balancer 🡪 entire world

Drawbacks with docker containers

Single host 🡪 suppose we have deployed some 100s of container on top of one single host, then due to some other container acquiring more space memory the other container die it won’t come up automatically

Auto healing 🡪 it’s not there in docker 🡪 auto healing means due to some issue that container which went down it has to come up automatically without users intervention

Auto scaling 🡪 this also not present in docker 🡪 if due to some festival number of users visiting to some site will be suddenly increasing then we have to scale up containers manually or automatically n distribute the load on all the containers equally using load balancer

Docker is simple it won’t provide enterprise level support(auto scale, firewall, api gateway, auto heal)

Docker vs Kubernetes diff

Kubernetes 🡪 Cluster 🡪 group of nodes 🡪 master n worker nodes 🡪 if one app or pod going down it will transfer it immediately on another node

Auto scaling 🡪 replica set or replica controller 🡪 if load is increasing then we have yaml which is an indentiation file in that we have add increase containers from 1 to 10

Auto healing 🡪 if some pod or container going down api server will receives a signal so it will launch new pod before terminating old one

Docker 🡪 container have container runtime 🡪 dockerstim

Request goes through control plane 🡪 master 🡪 API server exposes kuberntes to external world 🡪 scheduker reposible for scheduling pods or resources on kubernetes 🡪 whos decides info –api server, n who act’s on info that is kubelet 🡪 etcd backing store of entire cluster, info in form of key value pair 🡪 controller manager suppose take replica set if we have two pods so it will make sure that those two pods are always running, many controllers are there like replica set they make sure that some resources like pods they were always running 🡪 cloud controller manager if suppose we get some request to create storage then kubernetes has to translate that request into api request which cloud provider understands

Data plane 🡪 Worker node 🡪 kubectl it is responsible making sure pod is running if it’s not running there is auto healing in kubernetes it will take care of that 🡪 it have container runtime which runs container 🡪 kube proxy is provides networking 🡪 we have some containers n ip address will be allocated to it n provided load balancing also because kubernetes has auto scaling feature if you scale up the pod if you have two replicas instead of one then the load should be distributed 50-50 that is done by kube proxy

Pod is just like having some wrapper over container which has some advanced capabilities

Which kubernetes distribution you managed in production

What is distribution 🡪 we can consider eks as an distribution of kubernetes that is provided by amazon which has some additional wrappers which has some additional plugins n command line options

If you were using kubernetes then you won’t get support from amazon but if you will use eks then you will get support from amazon

Kops 🡪 kubernetes operations 🡪 we have to deal with installation, deletion n modifications of clusters which is called as life cycle of kubernetes which is managed by kops, it is most widely used for installing kubernetes

what is pod why we should deploy you container as a pod why can’t you directly deploy as a container in kubernetes 🡪 deploying your container as a pod in Kubernetes provides a more flexible, scalable, and manageable deployment model compared to deploying as a standalone container.

Pod 🡪 single container 🡪 here instead of running docker run n all those commands we have put all command in yaml file n we can execute that

When you put 2 containers in a single pod then kubernetes will provide some advantages shared storage shared networking n containers can communicate using localhost

Pod is wrapper created by kubernetes for containers suppose we have hundreds of containers then we can define everything about that in single yaml file which is easy to understand to dev also

Kubectl 🡪 command line for kubernetes 🡪 so we can directly interact with kubernetes clusters

Pod is described as a definition of how to run container

Two go to commands

Kubectl describe pod nginx 🡪 debug

Kubectl logs pod nginx

Steps

Create ec2 instance with 2 cpus

Install dependencies

Python3

AWS cli

Kubectl

Installing kops curl -LO https://github.com/kubernetes/kops/releases/download/$(curl -s https://api.github.com/repos/kubernetes/kops/releases/latest | grep tag\_name | cut -d '"' -f 4)/kops-linux-amd64

chmod +x kops-linux-amd64

sudo mv kops-linux-amd64 /usr/local/bin/kops

Set up aws configuration using aws configure

Kubernetes cluster installation

Create s3 bucket

aws s3api create-bucket --bucket kops-abhi-storage --region us-east-1

Create cluster

kops create cluster --name=demok8scluster.k8s.local --state=s3://kops-abhi-storage --zones=us-east-1a --node-count=1 --node-size=t2.micro --master-size=t2.micro --master-volume-size=8 --node-volume-size=8

kops edit cluster myfirstcluster.k8s.local

build cluster

kops update cluster demok8scluster.k8s.local --yes --state=s3://kops-abhi-storage

validate cluster

kops validate cluster demok8scluster.k8s.local

Deploy first app deployment using pod

First install pytho3, aws cli, kubectl

Kubectl version --client

Install docker

Sudo apt install docker.io

Sudo systemctl start docker

Sudo systemctl enable docker

sudo usermod -aG docker $USER

sudo usermod -aG docker $USER && newgrp docker

Install minikube(allow you to create kubernetes cluster)

curl -LO <https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64>

 sudo install minikube-linux-amd64 /usr/local/bin/minikube

minikube start

kubectl get nodes

pod installation

pod.yml 🡪 <https://kubernetes.io/docs/concepts/workloads/pods/>

kubectl create -f pod.yml

kubectl get pods

kubectl get pods –o wide

login to cluster minikube ssh

curl ip

container vs pod vs deploy

container 🡪 using containerized platform like docker we created container n we provide specification or command to run this container on command line

pod 🡪 instead of running commands on command line we have provide all those specification(port,image,volume) in yml file, in pod we can create single or multiple containers

pod is just providing yml specification for running container n we can run multiple container n those container can share storage netwokring, but we can’t achieve auto healing n auto scaling using pod

deployment 🡪 achieve auto healing n auto scaling using deploy n deploy it as a pod

don’t create pod directly create it using deployment resource n this deployment resource first create replica set which is a kubernetes controller n later it will create pod

If suppose in deployment yml file you added as you want two replicas of pod you can consider it as high availability then it will create two pods but this replica set it works as auto healing it will ensure that two pods are running if someone will delete one pod then it will again bring up another pod

To list out all the resources available in namespaces

Kubectl get all

Kubectl get all –A

vi deployment.yml 🡪 <https://kubernetes.io/docs/concepts/workloads/controllers/deployment/>

kubectl apply -f deployment.yml

kubectl get deploy

kubectl get pods

kubectl get rs

kubectl delete pod podname

kubectl get pods –w

before service 🡪 suppose we have some 30 users n we have three pods using auto healing if any pod goes down it will bring up automatically n the ip address will change so users who were accessing those ip they won’t able to access

after service 🡪 on top of deployment we can create service it will work as load balancer it will equally divide no. of users accessing ip but another issue it’s still there what of ip will change how user can access that app so for that service discovery(labels & selectors) came into picture 🡪 all people were assigned with same label if some pod will go down n ip will change that doesn’t matter because now it will recognize by label name

Cluser IP 🡪 load balancing, discovery, only accessed for those who are having access to kubernetes cluster

Nodeport 🡪 whoever has access to nodes they can access app, inside org

Load balancer 🡪 service expose app to external world

Enterprise & TLS load balancer

Load balancing type service is good but missing these features 🡪 Sticky, TLS, path, host, ratio based

Load balancer 🡪 cloud provider will charge for each n every service

Using load balancing service if you want to create 200-300 services then we have to create those many static ip address n we have pay to cloud provider for each ip

So we want only one ip we can achieve that by using ingress

When we care ingress resource then we have to create ingress controller also which will look for the ingress resource n it will simply update or configure load balancer

RBAC – Role based access controller

Two types users n service accounts

Users -> if suppose we have dev n qe engineers so based on their role we have to give access

Service aacounts 🡪 managing service access

Managing rbac 🡪 service account/users, roles/cluster role, role binding/crb

User management 🡪 offload um to identity providers

Roles – creating yml file n adding in that give access to secrets configuration files that is role n aatching these roles to users that is role binding

If we are creating role in namespace then it is role n if you are creating role within that scope of cluster that is cluster

For some advanced features we can create custome rerources

CRD  
CC

CR