**DevOps Interview question and answer:**

Why we are using a CI/CD pipeline?

CI/CD pipelines minimize the risk of errors by automating critical tasks such as building, testing, and deployment.

**Git:**

**Git is a version control system that allows developers to track changes in their code locally.**   
  
**Github:**

1. Why we are using GitHub?

Store the application code in the remote repository.

1. What are the stages in Git and GitHub?

* Working area
* Staging area
* Local repository
* Remote repository

1. What are the commands we are using Git and GitHub?  
   git init 🡪 to create a local repo

git add 🡪 add the application code working area to the staging area.

git commit -m “message”🡪 add the application code staging area to the local area.

git remote add origin (GitHub URL) 🡪 how to add the remote repo

git push origin branch name🡪 add the application code local repo to the remote repo.

git status 🡪 Check the status of the code to see what changes have occurred in the code.

git diff – to view the difference before moving to the staging area.

git log 🡪 view the commit log who did this?

git log –one line 🡪 view the commit log one line.

git show <commit\_id> 🡪 Uses for which codes are added.

git checkout (branch) 🡪 how to change the branch to branch

git pull (GitHub URL) 🡪 how to pull the remote repository code to your local repo.

**Jenkins:**

Jenkins is used to build and test your software projects continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build.

1. What are the plugins you are using?

Sonarscanner

Eclipse

Role-based strategy

Prometheus metric

Docker

Ansible

1. Why we are using a configured system?

It's used for configuring your tool in the configure system.

1. Why we are using a tool configuration?

Jenkins is extremely popular in the field of configuration management. For example, the installation service needs to be configured to the tool's configuration.

1. Why we are using a credential?

secret text, username, and password, as well as secret file credentials. For example, your credentials can be used securely in an encrypted format.

1. How to add the user in Jenkins?

Using manage user

1. How to change the Jenkins port number?

cd /etc/default/

vi jenkins

HTTP\_PORT:9999

1. How to verify Jenkins which port running?

ps -ef | grep jenkins

1. What is Jenkins default path?

cd /var/lib/Jenkins

1. How to identify the Jenkins job in the server?

cd /var/lib/Jenkins/workspace

1. How to start your Jenkins server?

service jenkins start

1. How to stop your Jenkins server?

service jenkins stop

1. How to stop your Jenkins server?

service jenkins status

**Ansible:**

1. Why we are using Ansible?

Ansible lets you quickly and easily deploy multi-tier apps. You won't need to write custom code to automate your systems; you list the tasks required to be done by writing a playbook.

1. What is an inventory file?

You will store your slave's private IP in an inventory file.

Path: /etc/ansible/hosts

1. What is a configuration file?

Ansible configuration files details

/etc/ansible/ansible.cfg

1. What are adhoc commands?

ad hoc tasks can be used to reboot servers, copy files, manage packages and users, and much more.

ansible all -a ping

ansible sai -i slaves.txt -m yum -a "name=httpd state=present" -b

1. Why we are using roles?

Ansible roles allow you to develop reusable automation components by grouping and encapsulating related automation artifacts, like configuration files, templates, tasks, and handlers

1. Why are we using vault?

Ansible Vault is an Ansible feature that helps you encrypt confidential information without compromising security.

ansible-vault encrypt tomcat.yaml

password:

ansible-playbook -i slaves.txt tomcat.yaml --ask-vault-pass

password:

ansible-vault view tomcat.yaml

password:

**Terraform:**

You can create your infrastructure. HashiCorp Terraform is an infrastructure as code tool that lets you define both cloud and on-prem resources in human-readable configuration files that you can version, reuse, and share.

**Terraform and ansible difference:**

1. Ansible 🡪 not checking state file
2. Terraform 🡪 state file checking
3. Why we are using a version file?

mention your terraform version like 0.13

1. Why we are using a provider file?

You should provide the access key and secret key in provider file.

1. Why we are using a resources file?

You should mention the infrastructure details in the resource file. For example, EC2 creation, and VPC.

1. Why we are using a variable file?

Terraform variables are used to customize modules and resources without altering their source code.

1. What the terraform commands is there?

terraform init

terraform validate - validate your code (any error or not)

terraform fmt – Alignment

terraform plan - lets you preview the actions Terraform would take to modify your

infrastructure or save a speculative plan that you can apply later(what happens)

terraform apply - performs a plan just like terraform plan does, but then actually carries out the planned changes to each resource using the relevant infrastructure provider's API.

cat terraform.tfstate - actions happen.

terraform apply --auto-approve

1. What are the modules your using in your project?

VPC

EC2

CLOUDFRONT

CERTIFICATE MANAGER

S3

**Docker:**

1. Why we are using docker?

portability, ease of deployment, and smaller size compared to virtual machines. You can deploy the multiple application service in docker.

1. How to build your Docker images?

docker build -t (images name) .

1. How to run your Docker container?

docker run -itd --name containe name -p 8080:80 image name

1. How to remove your docker images?

docker rmi (image name)

1. How to stop your docker conatiner?

docker stop (container ID)

1. How to list the docker container?

docker ps -a

1. How to back up your container?

docker run -d -it --name=damo --monut source=my-vol,destination=/usr/local/apache2/htdocs httpd:2.4

1. What the things in Docker file?

FROM ubuntu

MAINTAINER hari

RUN apt-get update

RUN apt-get install vim git wget openssh-server -y

RUN sed -i 's/PermitRootLogin prohibit-password/PermitRootLogin yes/' /etc/ssh/sshd\_config

RUN echo 'root:root123' | chpasswd

RUN mkdir /var/run/sshd

CMD ["/usr/sbin/sshd", "-D"]

EXPOSE 22

1. How to tag the images?

docker tag centos image:latest

1. How to go to the docker container?

docker exec -it (ID) /bin/bash

1. How to push the docker image?

docker push images

1. How to view the container logs?

docker log container ID

**Kubernetes:**

Why we are using Kubernetes?

No down time

Auto-scaling

Self-healing

- A software tool to run Dockerized application in the Cluster of Nodes

- Used to Monitor & efficiently managing the environment of the Enterprise Application

- Also been called as Orchestrational Engine

- It take cares about the Stability of the Application, in order to brings up the functional units automatically.

**K8s Architecture: -**

- It's a management Architecture, that comprises various managable components into it.

- Which gives an Orchestration system

**- Master Node**

1) API server - Its a front end, to communicate & maintain configurations of Cluster

2) Controller Manager - Controls the number of POD replicas & creations.

3) Scheduler - Schedules the Pod activities.

4) etcd - Distributed key value db. All the details of the cluster will be stored & kept here.

**- Worker Node**

1) Kubelet - It is an agent/connectivity, to Monitor/ manage the containers in the Pod.

2) Kubeproxy - Bridge between App\_user & Application. It is a logical endpoint.

3) POD - Grouped containers, functional units in K8S.

4) Containers - Container runtime, K8S suppports Docker, Mesos, OpenVZ, Marathon.

**- Kubectl**

1) Its a tool used to communicate with the Kubernetes cluster.

2) It connected with the APIServer for the communication

3) ~/.kube/config file contains the cluster information and shares to kubectl

4) kubectl command format,

kubectl <Operation\_command> <Type\_command> <Name\_command>

- Operations:- Get, Create, Delete, Describe, Logs, etc

- Type:- Pods, Deployments, Jobs, Namespace, etc

- Name:- Seach-pod, Cart-Deployment, Ui-Service, mail-jobs, etc

eg:- kubectl get pod test-pod