Devops Interview Questions

Introduction :

I have overall 3+ years of experience in devops. I have good experience in version control system GIT and Build automation tool Maven, continuous integration tool jenkins, configuration management tools ansible and container Technology docker, container orchestration tool kubernetes and along with this i do have experience in cloud technology AWS and cloud automation tool terra form. These are the tools and technologies I have worked on in the last 4 years.

1. Terraform version ?
   Ans: 0.12

2.Tasks in Jenkins ?
Ans: 1.Configuring OS CI/CD
2.Creating CI/CD pipeline

3.Jenkins file?
Ans: I used a declarative pipeline. In this pipeline different stages are there.
The very first stage is clone stage
Source code is checkout from the repository
Maven goals:
- Clean
- Validate
- Compile
- Test
- Package
- Install
- Replay

4.How to backup Jenkins ?
Ans: Goto Jenkins home directory and taking as a bar backup.
From the main menu select Manage Jenkins, then go to Plugins>Available and search for backup.

5.Different plugins used in Jenkins ?
Ans: 1. Nexus artifact uploader
2.Maven installation
3.sonarqube analysis
4.Git customized folder
5.Git hub parameter
6.Copy SSHQ
7.Publish over SSH
8.Deploy to Container
9.Deploy to EBS

6. Why we are using Docker?
Ans: Basically docker is a lightweight container things.What ever the things we need to required to run an application all those are bundled together. That bundled we run an image container will be created.enables you to separate your applications from your infrastructure so you can deliver software quickly.

7.What is stateless and stateful applications in the docker ?
Ans: So basically statefull is it will store the data and it will persistence. The stateless is doesn't persistence the data so whatever the application properties and application things the data will be lost.

8. What is Git Stash?
Ans:Stash like you know whenever you have modified something in your work space if you don't want to commit those things at this point of time if you want to store these in a temporary area in a same memory location. At that scenario 2 have go with stash.

9.What is the master slave architecture in Jenkins?
Ans:Master slave architecture like you know in order to distribute our jobs to the multiple slaves in that case we have to use slaves. So basically in order to configure the slave machines and we need to run our jobs on the slave Machines.

10.What are the best strategies to write a docker file?
Ans:Keep the Dockerfile simple: A Dockerfile should be easy to read and understand. It should not contain unnecessary commands or steps.

Use a lightweight base image: Start your Dockerfile with a lightweight base image to reduce the image size and improve performance.
Run only one process per container: Each container should run only one process. This makes it easier to manage and debug the container.

Use COPY instead of ADD: Use COPY instead of ADD to copy files to the container. COPY is more explicit and has fewer side effects.

11.What are the different stages in CI/CD pipeline?
Ans: The different stages in CI/CD pipeline is

1.Clone Stage	2.Build Stage
3.Sonar analysis stage	4.Package stage
5.Docker image creation stage	6.Deployment stage

12.AWS what are the services you worked?
Ans:In AWS i worked on the services like you know EC2,S3, Route S3, Auto Scating, load balancer, RDS, SNS, SQS.These are the services which i have automated using terraform.

13. In Terraform how to create a variable?
Ans:In Terraform module there variable TF file is there.In that variable. TF file we have to declare a variable when we are calling the module we have to pass that value.

*14.Tell me about the terraform lifecycle?
Ans:Terraform init :Prepare your working directory for other comments.
Terraform validate: Check whether the configuration is valid.
Terraform plan: Show changes required by the current configuration.
Terraform apply:Create or update infrastructure.
Terraform destroy: Destroy previously created infrastructure.

15.What is the use of Docker file?
Ans: When we run that docker file the image will be created. So in that docker file there are some instructions are there.FROM, RUN, CMD, WORKDIP, ENTRY POINT,USER, EXPOSE, VOLUME.So these are the instructions which are there in the docker file.

16.Can you Explain what is POD ?what is the use of POD?
Ans: Pod is the minimal object which we can able to create in the kubernetes so inside the pod containers will be there.So that pod will contain one or more then one container.

17.How to delete the POD?
Ans: Kunbectl delete pod name -N name space.

18.How to merge 2 branches in Git ?
Ans: By creating the pull request.
If you want to merge something the branch will be pushed to a centralized repository.They will create a pull request in the github and they will put the reviewers who are part of that project. So they will reviewer, We were following 2 reviews once the 2 people were reviewed . Once they approve . Then we can able to merge our things into whatever branch to merge then the will merge.

19.what is the difference between rebase and merge in Git?
Ans: Rebase flattens the history, removing unwanted entries. Git merge, on the other hand, only changes the target branch and creates a commit, preserving the history of the source branch.
`

20.What are the Quality Gates in Sonarqube?
Ans: In SonarQube, quality gates are a set of predefined rules or conditions that determine whether a project's code quality meets certain criteria. Quality gates are used to ensure that code quality standards are maintained

throughout the software development process, and to prevent low-quality code from being released into production.

21.What is VPC ?What are the components of VPC ?
Ans:VPC is a virtual private cloud. Which is an isolated network .aZ
The VPC Components are

1. Internet gateway 2.Route tables 3.subnets
4.Net gateway 5.VPC peering 6.Elastic IP addresses

These are the major components in the VPC

22.What is internet gateway in VPC?
Ans:The internet gateway that allows communication between your VPC and the internet.

23.Can you explain what are the task perform in Jenkins ?
Ans: Jenkins is a popular open-source automation server used for continuous integration and continuous delivery (CI/CD) of software applications. It provides a variety of features and plugins that allow developers and operations teams to automate different stages of the software development lifecycle. Here are some common tasks performed in Jenkins:
Building code, Continuous Integration (CI), Continuous Delivery (CD), Code analysis and testing, Reporting, Plugin management, Automation.

24.What is your Experience in Kubernetes ?
Ans:In kubernetes i do have experience in the setting of the kubernetes in kubespray incubator project and then i do have experience in like you know deploying the API objects in the kubernotes and then i do have experience in the upgrading the kubernetes from one version to another version using kube spray in the on frame.

25.Which version of kube spray which you have used?
Ans: so 2.23 which i used which i used that kube spray. I will get you know 1.11 version of kubernetes.

26. Why do we need kubernetes?
Ans: Upgrading the cluster and day to day base our docker image we will deploying into the kubernetes cluster.

27.What do you feel about the devops ?
Ans: Devops basically development of operations. Devops is the person has to coordinate with dev team and ops team and then make sure he has what ever the things which is the deliverable he has to do it in that in time.

28.Can you tell me about the branching strategies in GIT ?
Ans: So branching strategy which we followed like you know we do have feature branch and release branch.whatever the code which is in the main

branch the code will be deployed into the production environment and whatever the developers working on feature branch they will work on those things. Once whatever the features which they developed, once there features working properly then they make sure the dev branch .Once the dev branch code deployed into the development environment once it's signouts the code will make sure to the release branch.

29.Can you explain life cycle of Mavan?
Ans: There are 3 different life cycles available in maven

1.Define life cycle
2.Clean life cycle
3.site life cycle.

In default life cycle different stages are available.
1.Validate 2.compile 3.Test compile 4.package 5.Install 6.Deploy

30.How ansible will work?
Ans:Ansible is basically a configuration management tool.It work like you know 22 protocol. Basically it connecting to the linux machines using on SSH protocol. Ansible internal uses python so whatever the modules which are there in ansible those modules developed by python.

31.Tell me 5 commands you used in Linux ?
Ans:1.mkdir Command. To create folders in the shell, you use the mkdir command.
2.The pwd command is used to display the location of the current working directory.
3.cat command. Concatenate, or cat, is one of the most frequently used Linux commands.
4.touch command. The touch command allows you to create an empty file.
5.chmod is a common command that modifies a file.

32.How to find the kernel version in Linux ?
Ans:To check Linux Kernel version, try the following commands: uname -r.

33.What is VPC ? and how will it work ?
Ans:VPC is a virtual private cloud. It is an isolated network across your account. So in order to configure the customnetwork you need to define on VPC.

34.Tell me what you did in docker ?
Ans: So docker file creation and building the docker images and creating the containers. i am take care of containers or any thing like you know whatever the containerization application which is there so i have migrated those applications to the kubernetes cluster.

35.How to copy files from host to docker container?
Ans:Using docker CP command.

36.Kubectl authentication and authorization?
Ans:In kubectl wherever you can install the kubectl in the home directory you should have one file context file using that context file you can able to admit to the kubernetes API server authentication and authorization.Authentication means whatever the having on the cluster and anthourized like is it deployment or is it pod.

—----------

1. Can you tell something about docker container?

- In simplest terms, docker containers consist of applications and all their dependencies.
- They share the kernel and system resources with other containers and run as isolated systems in the host operating system.
- The main aim of docker containers is to get rid of the infrastructure dependency while deploying and running applications. This means that any containerized application can run on any platform irrespective of the infrastructure being used beneath.
- Technically, they are just the runtime instances of docker images.

2. What are docker images?

They are executable packages(bundled with application code & dependencies, software packages, etc.) for the purpose of creating containers. Docker images can be deployed to any docker environment and the containers can be spun up there to run the application.

3. What is a DockerFile?

- It is a text file that has all commands which need to be run for building a given image.

4. Can you tell what is the functionality of a hypervisor?

A hypervisor is a software that makes virtualization happen because of which is sometimes referred to as the Virtual Machine Monitor. This divides the resources of the host system and allocates them to each guest environment installed.

- This means that multiple OS can be installed on a single host system. Hypervisors are of 2 types:

1. **Native Hypervisor:** This type is also called a Bare-metal Hypervisor and runs directly on the underlying host system which also ensures direct access to the host hardware which is why it does not require base OS.
2. **Hosted Hypervisor:** This type makes use of the underlying host operating system which has the existing OS installed.

5. What can you tell about Docker Compose?

It is a YAML file consisting of all the details regarding various services, networks, and volumes that are needed for setting up the Docker-based application. So, docker-compose is used for creating multiple containers, host them and establish communication between them. For the purpose of communication amongst the containers, ports are exposed by each and every container.

6. Can you tell something about docker namespace?

A namespace is basically a Linux feature that ensures OS resources partition in a mutually exclusive manner. This forms the core concept behind containerization as namespaces introduce a layer of isolation amongst the containers. In docker, the namespaces ensure that the containers are portable and they don't affect the underlying host. Examples for namespace types that are currently being supported by Docker – PID, Mount, User, Network, IPC.

7. What is the docker command that lists the status of all docker containers?

In order to get the status of all the containers, we run the below command: `docker ps -a`

8. On what circumstances will you lose data stored in a container?

The data of a container remains in it until and unless you delete the container.

9. What is docker image registry?

- A Docker image registry, in simple terms, is an area where the docker images are stored. Instead of converting the applications to containers each and every time, a developer can directly use the images stored in the registry.
- This image registry can either be public or private and Docker hub is the most popular and famous public registry available.

10. How many Docker components are there?

There are three docker components, they are - Docker Client, Docker Host, and Docker Registry.

- **Docker Client:** This component performs “build” and “run” operations for the purpose of opening communication with the docker host.

- **Docker Host:** This component has the main docker daemon and hosts containers and their associated images. The daemon establishes a connection with the docker registry.
- **Docker Registry:** This component stores the docker images. There can be a public registry or a private one. The most famous public registries are Docker Hub and Docker Cloud.

11. What is a Docker Hub?

- It is a public cloud-based registry provided by Docker for storing public images of the containers along with the provision of finding and sharing them.
- The images can be pushed to Docker Hub through the `docker push` command.

12. What command can you run to export a docker image as an archive?

This can be done using the docker save command and the syntax is: `docker save -o <exported_name>.tar <container-name>`

13. What command can be run to import a pre-exported Docker image into another Docker host?

This can be done using the docker load command and the syntax is `docker load -i <export_image_name>.tar`

14. Can a paused container be removed from Docker?

No, it is not possible! A container MUST be in the stopped state before we can remove it.

15. What command is used to check for the version of docker client and server?

- The command used to get all version information of the client and server is the `docker version.`
- To get only the server version details, we can run `docker version --format '{{.Server.Version}}'`

—----------

DOCKER:

WHY DOCKER: docker ensures that your application will be same on your machine and any other machine in the world.

DOCKER DAEMON: it receives the commands from the clients and manages the objects like images, containers, networks and volumes.

CONTAINER:

It is a way of packing an application with all the necessary dependencies and configuration files.

It a directory which has all the libraries/binaries to run the service.

directory has its own IP address and port numbers to access the services.

Container is a virtual machine which doesn't have any OS because it shares the host machine OS system.

Docker containers are standard, security and lightweight.

Docker engine combines the namespaces(name of the container), Cgroups(CPU, Memory & networks) and UnionFS(File systems) to form a container.

IMAGES:

Docker images are used to pack the application.

It consists of multiple layers. Each layer will be some set of data. If you add some files it will add a layer to the image, when you change any configuration files that will create a layer.

When we create image from the container. That image will be read only mode.

Images becomes containers when we run it.

When you add some data in a container and commit it will automatically create a layer.

Docker exec - used to run any command in the container from the outside of the container

Docker it - interactive terminal, if we are running a container without 'it' container will be exited state because it runs on bin/bash command, due to the absence of 'it' the shell was dead and container also dead.

ctrl+p+q ——> used to exit from the container without stopping the container.

==========

RUN: it is used to execute the commands while we build the images and add a new layer into the image.

```
FROM centos:centos7
RUN yum install git -y
```

or

```
RUN ["yum", "install", "git" "-y"]
```

CMD: it is used to execute the commands when we run the container.

It is used to set the default command.

if we have multiple CMD's only last one will gets executed.

```
FROM centos:centos7
CMD yum install maven -y
```

or
CMD [“yum”, “install”, “maven”, “-y”]

If you want to overwrite the parameters:
docker run image_name httpd (FAILED)
docker run image_name yum install httpd -y (only httpd will gets installed)

ENTRYPOINT: it overwrites the CMD when you pass additional parameters while running the container.

FROM centos:centos7
ENTRYPOINT [“yum”, “install”, “maven”, “-y”]

If you want to overwrite the parameters:
docker run image_name httpd (both maven and httpd will gets installed)
docker run image_name yum install httpd -y (both maven and httpd will gets installed)

FROM centos:centos7
ENTRYPOINT [“yum”, “install”, “-y”]
CMD [“httpd”]

Bydefault it will executes httpd command, if you specify the command while running the container it will gets executed.
docker run image_name (httpd will install)
docker run image_name git (only git will install)
docker run image_name git tree(both git & tree will install)

COPY: It copies the files from local server to container.
ADD: It will also copies the files but the difference is it can handle remote URLs & it can auto-extract tar files.
ARG: it is used to define a variable that is used to build a docker image, it will not available once we build it. In containers also it’s not possible to access it.
we can change these variables in command line arguments (docker build -t test --build-arg var_name=value)
ENV: These variables are used for inside the container
we can’t change the values in command line arguments. f we need to change the ENV variable using CMD line then we have to use ARG and place ARG variable in ENV variable

ARG abc=devops
ENV xyz=aws
RUN echo $abc

RUN echo $xyz
==========

DOCKER NETWORK: it allows you to attach your container into many networks, it is used to isolate the containers

TYPES:
* Bridge: This is the default network driver and is used to create a standalone network for containers running on a single host. Containers on the same bridge network can communicate with each other directly and can also access external networks through the host.
* Host: This network driver removes network isolation between the container and the host, allowing the container to directly access the host's network stack and all its connected networks.
* Overlay: This network driver enables multi-host networking and is used to connect containers running on multiple Docker hosts. This allows containers to communicate with each other across hosts as if they were on the same network.
* Macvlan: This network driver allows a Docker container to have its own unique MAC address, effectively giving it a directly routable IP address. This is useful for scenarios where a container needs to appear as a physical device on the network.

To create a network: docker network create —attachable network_name
To see the list: docker network ls
To delete a network: docker network rm network_name
To inspect: docker network inspect network_name
To connect a container to the network: docker network connect network_name container_id/name
To disconnect from the container: docker network disconnect network_name container_name
To prune: docker network prune
==========

Docker file for springboot application:

From openjdk:8-jdk-alpine
Copy target/*.jar code/
Entyrpoint [“java”, “-jar”, “jar_filename”]

==========

DOCKER RENAME:
To rename docker container: Docker rename old_container new_container
To rename docker port:

- stop the container
- go to path (var/lib/docker/container/container_id)
- open hostconfig.json
- edit port number
- restart docker and start container

==========

DOCKER SECRETS: by using this docker secrets, we securely store data that can be read by applications at runtime. his includes data such as passwords, hostnames, SSH keys, and more.

let's say our application requires a database connection. To do this, it needs a hostname, username, and password.
Furthermore, there's a different database server for development, testing, and production.
With docker secrets, each environment can provide its own database information to the applications.

Create a file which contains sensitive content: vim docker/password/secrets/file1
Username: Mustafa
password: 1234
Hostname: 111.23.45.2

To create a secret: docker secret create secret_name path(docker/password/secrets/file1)
To inspect a secret: docker secret inspect secret_name
To se the list: docker secret ls
To remove a secret: docker secret rm secret_name
docker service create --name my_app --secret mustafa openjdk:19-jdk-alpine

==========

DOCKER EXPORT: It is used to save the docker container to a tar file
Create a file which contains will gets stored: touch docker/password/secrets/file1.txt
TO EXPORT: docker export -o docker/password/secrets/file1.txt container_name
SYNTAX: docker export -o path container

==========

DOCKER SWARM: It is a group of servers that runs the docker application. This can be implemented by the cluster.
The activities of the cluster are controlled by a swarm manager, and machines that have joined the cluster is called swarm worker.
It is used to manage the containers on multiple servers

To create a service: docker service create —name devops —replicas 2 image_name
Note: image should be present on all the servers
To update the image service: docker service update —image image_name service_name
Note: we can change image,
To rollback the service: docker service rollback service_name
To scale: docker service scale service_name=3
To check the history: docker service logs
To check the containers: docker service ps service_name
To inspect: docker service inspect service_name
To remove: docker service rm service_name

==========
SOME EXTRA POINTS:
docker run -itd —rm —name Mustafa ubuntu - if you use —rm in a container, it will work since it Is in run state, when it goes to exited, it will gets deleted automatically
docker exec -it cont_name /bin/bash - used to enter into daemon containers

==========

```
version: "3.1"
services:
  mobile_recharge:
    image: nginx
    ports:
      - "9999:80"
    volumes:
      - "mobile-recharge-volume1"
    networks:
      - "mobile-recharge-network"

  mobile_recharge2:
    image: nginx
    ports:
```

```
      - "9998:80"
networks:
  mobile-recharge-network:
   driver: bridge
```

docker-compose up -d - used to run the docker file
docker-compose build - used to build the images
docker-compose down - remove the containers
docker-compose config - used to show the configurations of the compose file
docker-compose images - used to show the images of the file
docker-compose stop - stop the containers
docker-compose logs - used to show the log details of the file
docker-compose pause - to pause the containers
docker-compose unpause - to unpause the containers
docker-compose ps - to see the containers of the compose file

DOCKERIZING SPRING BOOT APPLICATION APP VS JAVA APPLICATION

SPRING BOOT:
- [ ] When we build the code, It will generates a jar file.
- [ ] It will have a embedded(internal) server, we don't need to configure separately.
- [ ] Command to run app: ["java", "-jar", "jar filename.jar"]

JAVA APP:
- [ ] When we build the code, It will generates WAR file
- [ ] It needs to configure a web server like tomcat to run the application.

DOCKER FILE FOR SPRING BOOT APPLICATION

```
FROM java:8-jdk-alpine
COPY target/app.jar /usr/app/app.jar
WORKDIR /usr/app
ENTRYPOINT ["java", "-jar", "app.jar"]
```

GIT:

1. What is git?
Git is nothing but Global Information Tracker.
Git is also called as version control system and source code management.
It is used to track the files.

It will maintain the multiple versions of the same file.
Git is free and open source.

2. What are git stages?

There are 3 stages in git which is
1. Working directory
2. Staging areas
3. Repository

Working directory: in this stage we can create a files for our project.

Staging area: it is nothing but a rough draft space
We can modified file in its current version to go into your next commit snapshot.
We can track the files in staging area.

Repository: it is nothing but out project folder, when we commit the file it will store on repository.
the data is safely stored in your local database.

We have 3 types of repositories
1. Local
2. Remote
3. Central

3. How to track a file in git?
To track a file, we use a command “git add filename”
Once we add, it will gets started tracking
So that we can know the every modification on the file

4. How to commit the file?
To commit a file we use a command called “git commit -m “message” file_name
Once we commit the file it will safely stored in local database.

5. What is git status?
Git status is used to know the tracking and untacking files in our server.

6. What is git logs?
Git log is a history, we can see the list of every commit in our project along some other information like who commit the file and when we commit the file.

7. How to configure in git
Git config user.name “name”
Git confit user.email “name@gmail.com”

8. What is git merge?
It allows us to get the code from one branch to another. This is useful when developers work on the same code and want to integrate their changes before pushing them up in a branch.

Command: git merge branch_name

9. What is merge conflicts?
Merge conflicts happen when you merge branches that have competing commits, and Git needs your help to decide which changes to incorporate in the final merge.

10. What is git diff?
It is used to show all the differences between any two commits or files within your Git repository.

11. What is git push?
Git push is used to share the local files into central repositories like GitHub and Bitbucket.

12. What is git stash?
It is used to save your changes but not record them in the Git repository.

13. What is git clone?
Git clone is used to get the repository from central to local.

14. What is Git checkout?
Git checkout is used to switch the branches

Command: git checkout branch_name

15. What is git rm?
Git rm is used to untrack the files

16. Difference b/w git and GitHub?

Git is a software	GitHub is a service
Git can be installed locally on the system	GitHub is hosted on the web
Git has 3 types of repositories	GitHub has 2 types of repositories
Git has a default branch called master	GitHub has a default branch called main

17. What is git pull?
Git pull is the combination of git fetch and git merge
Git pull is used to get the commits from central repo to local repo

Command: git pull origin branch

18. Git fetch vs pull

Git Fetch Git Pull
The Git fetch command only downloads new data from a remote repository.
Git pull updates the current HEAD branch with the latest changes from the remote server.
It does not integrate any of these new data into your working files.
Downloads new data and integrate it with the current working files.
Command - git fetch origin
git fetch --all Tries to merge remote changes with your local ones.
Command - git pull origin master

19. How to resolve the git conflicts?
* Open the file in vi editor and remove the conflict messages.
* Save the file
* Add the files using the git add command.
* The last step is to commit the changes in the file.

20. What is git tag?
Tagging is generally used to capture a point in history that is used for a marked version release.

21. How to change the commit message
git commit —amend commit_id

22. What is git fork?
Git fork is used to get the repositories from another account

23. What is cherry-pick?
Cherry-pick is used to get a particular file from another branch
command: git cherry-pick commit_id

24. What is git branch?
A branch is a way to isolate development work on a particular aspect of a project.

25. What is git blame?
It is used to see the history of the GitHub file.
==========

Git reset commit_id : deletes the commits
git log --graph --oneline --all : see all the history in graph
git merge --abort : cancel the conflicts in commits
git remote show origin : used to see the branches in GitHub

git log origin/main : history of the GitHub
git push --delete origin mustafa : delete the branch on GitHub
Git branch -m old new : to change the branch name

Extra commands:

git show <commit> --stat : you'll see the commit summary along with the files that changed and details on how they changed.
git commit --amend -m "New commit message" : to edit the commit message
git commit --amend --no-edit : used to add some files in previous commit. (--no-edit means that the commit message does not change.)
git update-ref -d HEAD : used to delete 1st commit in the git
git reset commit: used to delete all the commits (upto the commit id)

Let's say that you forgot to configure the email and already did your first commit. Amend can be used to change the author of your previous commit as well. The author of the commit can be changed using the following command:

git commit --amend --author "Author Name <Author Email>"

git merge vs git rebase
When you want to get the changes from one branch to another, we can do merge or rebase

Merging preserves the history of both branches, while rebasing rewrites the history of the branch being rebased.
Merging creates a new commit that has two parent commits, while rebasing applies the changes from one branch onto another.
Merging is generally easier and safer, while rebasing can be more complex and risky.

GITHUB
Git remote -v : used to see the remote repo
Git pull origin branch: used to get changes in GitHub
Git remote rename old -link new-link: change the repo
Git fetch origin master: to fetch from remote
Git merge origin/master : to merge from remote repo
git push origin --delete branch_name : to delete GitHub branch
Git fetch —all : used to fetch all the branches
git push -u origin --all : to push all branches
Git push -u origin branch1 branch2 : to push specific branches

—----------

Q. What is a GitHub?

A. GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere. This will teach you GitHub essentials like repositories, branches, commits, and Pull Requests.

Q. How to use GitHub?

A. Step 0: Install git and create a GitHub account. ...

Step 1: Create a local git repository. ...

Step 2: Add a new file to the repo. ...

Step 3: Add a file to the staging environment. ...

Step 4: Create a commit. ...

Step 5: Create a new branch.

step 6: Now push that branch to the repo

Q. Is GitHub open source?

A. Actually Git is not open source. Because it has Commercial offer known as "Git Enterprise Edition"

Anyone can apply their ideas like open source on GitHub alike known as Gitlab which is a Ruby

Application with its source code here.

Q. What is a Branch in GitHub?

A. A branch in Git is simply a lightweight movable pointer to one of these commits. The default branch name in Git is master. As you start making commits, you're given a master branch that points to the last commit you

made. Every time you commit, the master branch pointer moves forward automatically.

Q. What is a git Pull?

A. The git pull command is used to fetch and download content from a remote repository to local.

Q. What is meant by Git repository?

A. Git is a program that tracks changes made to files. ... A Git repository is the. git/ folder inside a project. This repository tracks all changes made to files in your project, building a history over time.

Q. What is meant by fork in GitHub?

A. Fork is a rough copy of the repository which allows you to test and debug the changes with changes and it does not affect the original code anymore.

Q. What is the difference between a fork and a branch in Git?

A. Branching and forking provide two ways of diverging from the main code line. ... So, unlike a branch, a fork is independent from the original repository. If the original repository is deleted, the fork remains. If you fork a repository, you get that repository and all of its branches.

Q. What is Git Fetch?

A. git fetch is the command that tells your local git to retrieve the latest meta-data info from the original.

Q. What is .gitignore file ?

A. if you want to ignore some specific files you can mention where and those files won't be tracked anymore.

Q. What is the difference between Master and Main Branch?

A. when you start extracting files and implementing it automatically you will get the main branch there so there you have to do all the activities merge, pull ... and eventually it all goes to the master branch.

Q. Which generation is the Git comes under?

A. Git comes under 3 rd. generation of VCS.

—----------

1) What is Jenkins and why should we use it?

Jenkins is a leading open-source, free automation tool that is used to develop and test software projects.

Reasons why Jenkins is widely used:

-Used to detect faults in software development and systematize the testing of buildings
-Used to constantly monitor the code simultaneously and add changes to the build.
-Jenkins consists of an installer package for most operating systems.
-Used to keep the team updated and synchronized about the changes incorporated.
-Used to build CI/CD pipelines since it has plugin capabilities and is easy to use.

2) What are the features of Jenkins?

Jenkins features include:

-It is an open-source automation tool and it is free.
-Gives pipeline support
-installation is easy on systems with multiple operations.
-large number of plugins
-Jenkins upgrades effortlessly.
-speedy release cycle
-Configuration setup is easy.

3) What are the prerequisites to use Jenkins?

The requirements to use Jenkins are:

-First requirement is an accessible source code repository, for instance, a Git repository.
-A build script in working condition, example- Maven script checked into the repository.

4) How do we manually restart Jenkins?

The following commands are used to restart Jenkins manually:
-(jenkins_url)/safe restart — Wait until all the builds are completed before restarting.
-(jenkins_url)/restart-To force restart without waiting for build completion.

5) What does “continuous integration” mean?
Continuous Integration is the continuous process of checking the code made by developers into a version control system numerous times. The build is automated in the process to inspect and detect bugs in the developed code. Continuous integration comprises of:

6) what is Jenkins Pipeline?
Jenkins Pipeline is collection of features of Jenkins. They are installed as plugins that allows delivery of pipeline implementation continuously.

7) What are the advantages of using Jenkins?

The advantages of using Jenkins are:

-User-friendly, free, and it is an open source
-Trouble-free Installation
-Code deployment is convenient and takes very less time. It simultaneously generates reports.
-Helps in collaboration between the operation and development teams.
-Free of cost
-Detection of code errors at an early stage
-Reduced integration issues due to automation of integration work
-Rich plugin ecosystem
-Platform independence

8) What are the components that you can integrate Jenkins with?

Jenkins is mainly integrated with the following components:
-First is the version control system, for example: GIT, SVN
-Another one is build tools. An example is Apache Maven.

9) How does Hudson relate to Jenkins?

Jenkins was previously known as Hudson. It is a continuous integration tool and is open source, written in Java.

10) Explain how to set up Build jobs in Jenkins.

The following steps will help you to build jobs in Jenkins:
Step 1 -First, open the Jenkins dashboard and click on the New Item.
Step 2 -Enter the item name and choose the “Freestyle project option”.
Step 3 -Specify the details of the job.
Step 4 -Next, specify the location of files that need to be built.
Step 5 -In case the repository is hosted on GitHub, enter the URL of that repository here.
Step 6 -Build section and click on the Add build step.
Step 7 -In the command window, enter the following commands and then click on the Save button.
(Javac HelloWorld.java
Java HelloWorld)
Step 8 -Now click on the “Build” option to see whether the job is defined well or not.
Step 9 -Once the build is scheduled, it will run.
Step 10 -To see details of the build, click on the Console Output link.

—----------

Ansible interview questions?

1) What Is Ansible?
Ansible is a configuration management system. It is used to set up and manage infrastructure and applications. It allows users to deploy and update applications using SSH, without needing to install an agent on a remote system.
2) What’s the use of Ansible?
Ansible is used for managing IT infrastructure and deploy software apps to remote nodes.
For example, Ansible allows you to deploy as an application to many nodes with one single command. However, for that, there is a need for some programming knowledge to understand the ansible scripts.
3) What is Ansible Galaxy?
Ansible can communicate with configured clients from the command line by using ansible command. It also allows you to automate configuration by using ansible-playbook command. To create the base directory structure, you can use a tool bundled with Ansible which is known as ansible-galaxy.
Command:

```
$ ansible-galaxy init azavea. packer
```

azavea.packer was created successfully

4) What is Continuous Delivery?
Continuous delivery is a practice of delivering the software as soon as it developed. In this method, we need to use versioning control system. The software is constantly updated in live production systems.

5) What is the way to access shell environment variables in Ansible?
In Ansible, if you want to access existing variables, the user needs to use the 'env' lookup plugin. Example, to access the value of the Office environment on the management machine:
You need to write following code:

```
---
# ...
vars:
local_home: "{{ lookup('env','Office') }}"
I
{{ ansible_env.SOME_VARIABLE }}
```

6) What is the code you need to write for accessing a variable name?
Variable names can be built by adding using the following method:

```
{{ hostvars[inventory_hostname]['ansible_' + which_interface]['ipv4']['address'] }}
```

The method of using hostvars is important because it's a dictionary of the entire namespace of variables. 'inventory_hostname' variable specifies the current host you are looking over in the host loop.

7) Explain how you can disable cowsay?
If cowsay is installed then executing playbooks inside the Ansible you can disable coway by using following options:

1. Uninstall cowsay
2. Setting up value for the environment variable

```
export ANSIBLE_NOCOWS=1
```

8) Explain how you can copy file recursively onto a target host?
The "copy" module has a recursive parameter. However, if you want this to perform more efficient for a large number of files, then "synchronize" module is the best option for you.

9) How Can you submit a change to the Documentation in Ansible?
Documentation for Ansible is kept in the project git repository. It contains complete instructions for contributing can be found in the docs.

10) What Is the Best Method to Make Content Reusable/redistributable?

You can read everything about “Roles” in the playbooks documentation section. This helps to make playbook content self-contained and shareable with other ansible users.

11) What is Ansible Tower?

Ansible tower is a tool which makes Ansible very easy to use. It acts as a hub for the task automation. The tower is free for usage till 10 nodes.

12) What’s the method to check the inventory vars defined for the host?

For that use this command:

```
ansible -m debug -a "var=hostvars['hostname']" localhost
```

Variable Name	Environment Variables
It can be built by adding strings.	To access the environment variable, you need to access existing variables.
{{ hostvars[inventory_hostname][‘ansible_’ + which_interface][‘ipv4’][‘address’] }}	# … vars: local_home: “{{ lookup(‘env’,’HOME’) }}”
Allows to add strings	To set environment variables, we need to see the advanced playbooks section.
Ipv4 address type use for Variable names we use the ipv4 address.	For Remote environment variables, use {{ ansible_env.SOME_VARIABLE }}

13) State the difference between Variable name and Environment Variables.

14) What are ad-hoc commands?

You can think of ad-hoc commands as a way for us to take actions on our hosts without writing a playbook. For example, if we want to reboot all hosts in a particular group(webservers). Then you can write a playbook or simply run a one-off ad-hoc command.

15) Explain Ansible facts?

You can think of ansible facts as a way for ansible to get information about a host and store them in variables for easy access. This information stored

in predefined variables are available to use in the playbook. To generate facts, ansible runs the setup module.

16) How do you see all variables for a host?

You can see them using the hostvars variable. This stores host variables with the hostname as key. For example, to look at the variables defined for localhost, you can run;

ansible -m debug -a "var=hostvars[inventory_hostname]"

17) Explain modules in ansible

Modules in Ansible are idempotent. From a RESTful service standpoint, for the operation to be idempotent, clients can perform the same result by using modules in Ansible. Multiple identical requests become a single request.

There are two different types of modules in Ansible:

- **Core modules**
- **Extras modules**

Core Modules

The Ansible team maintains these types of modules, and they will always ship with Ansible software. They will also give higher priority for all requests than those in the “extras” repos.

Extras Modules:

These modules currently is bundled with Ansible but might available separately in the future. They are also mostly maintained by the Ansible community. These modules are still usable, but it can receive a lower rate of response to issues and pull requests.

18) When should you test playbooks and roles?

In ansible, Tests can be added either in new Playbooks or to existing Playbooks. Therefore, most of the testing job offers a clean hosting each time. By using this testing methodology, you need to make very little to no code changes.

19) Discuss method to Create an Empty File with Ansible

To create and empty file you need to follow given steps.

Step 1. Save An Empty File into The Files Directory

Step 2. Copy It to The Remote Host.

You can think of ad-hoc commands as a way for us to take actions on our hosts without writing a playbook. For example, if we want to reboot all hosts in a particular group(webservers). Then you can write a playbook or simply run a one-off ad-hoc command.
