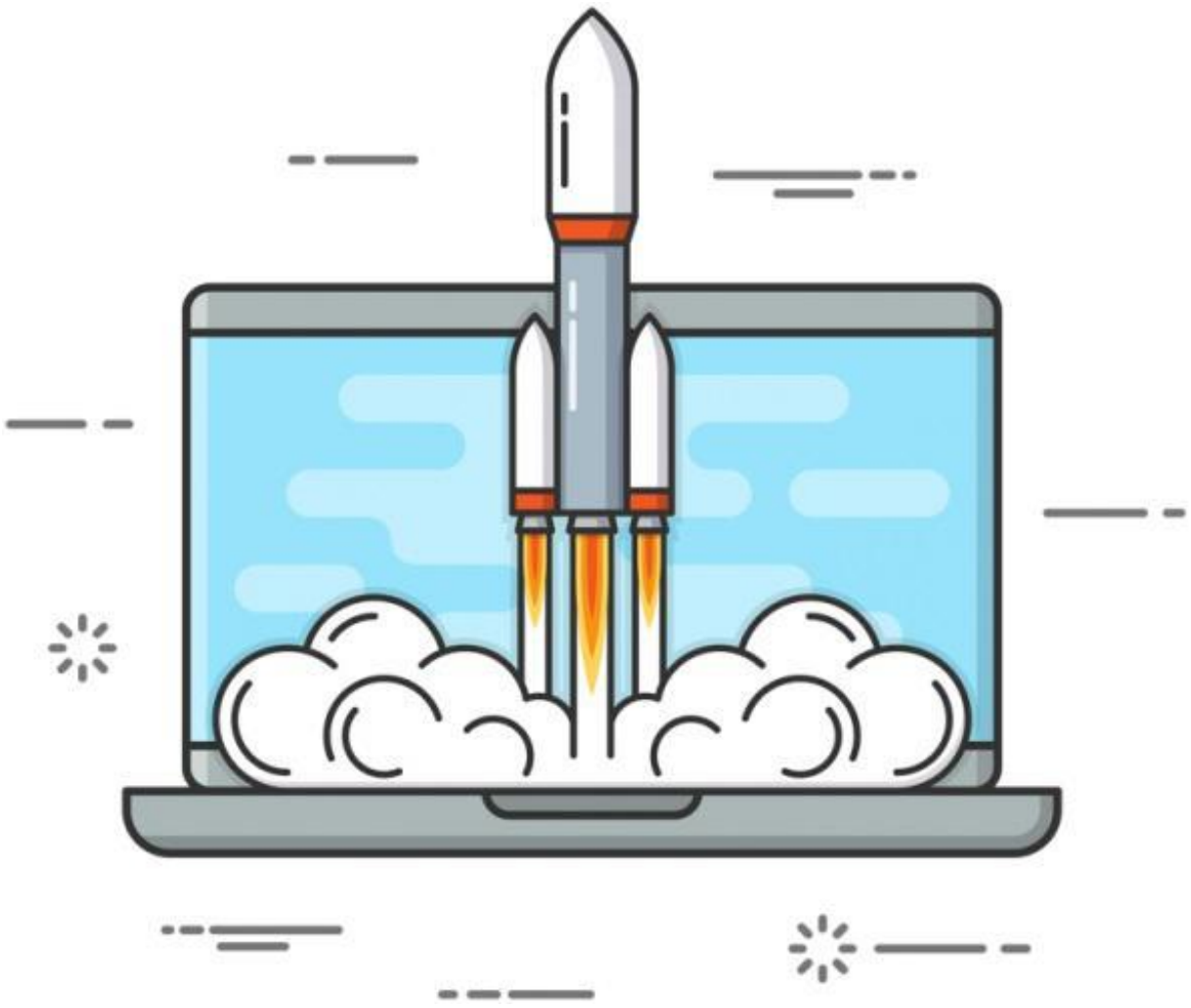


Everything About Devops

Thanks for the patience
pls share with friends

Q1) what is DevOps?

By the name DevOps, it's very clear that it's a collaboration of Development as well as Operations. But one should know that DevOps is not a tool, or a software or framework, DevOps is a Combination of Tools which helps for the automation of whole infrastructure.

DevOps is basically and implementation of Agile methodology on Development side as well as Operations side.

Q2) why do we need DevOps?

To fulfil the need of delivering more and faster and better application to meet more and more demands of users, we need DevOps. DevOps helps deployment to happen really fast compared to any other traditional tools.

Q3) Mention the key aspects or principle behind DevOps?

The key aspects or principle behind DevOps is:

Infrastructure as a Code

Continuous Integration

Continuous Deployment

Automation

Continuous Monitoring

Security

Q4) List out some of the popular tools for DevOps?

Git

Jenkins

Ansible

Puppet

Nagios

Docker

ELK (Elasticsearch, Logstash, Kibana)

Q5) what is a version control system?

Version Control System (VCS) is a software that helps software developers to work together and maintain a complete history of their work.

Some of the feature of VCS as follows:

Allow developers to wok simultaneously

Does not allow overwriting on each other changes.

Maintain the history of every version.

There are two types of Version Control Systems:

Central Version Control System, Ex: Git, Bitbucket

Distributed/Decentralized Version Control System, Ex: SVN

Q6) What is Git and explain the difference between Git and SVN?

Git is a source code management (SCM) tool which handles small as well as large projects with efficiency. It is basically used to store our repositories in remote server such as GitHub.

GIT	SVN
Git is a Decentralized Version Control Tool	SVN is a Centralized Version Control Tool
Git contains the local repo as well as the full history of the whole project on all the developers hard drive, so if there is a server outage , you can easily do recovery from your team mates local git repo.	SVN relies only on the central server to store all the versions of the project file
Push and pull operations are fast	Push and pull operations are slower compared to Git
It belongs to 3rd generation Version Control Tool	It belongs to 2nd generation Version Control tools
Client nodes can share the entire repositories on their local system	Version history is stored on server-side repository
Commits can be done offline too	Commits can be done only online
Work are shared automatically by commit	Nothing is shared automatically

Q7) what language is used in Git?

Git is written in C language, and since its written in C language its very fast and reduces the overhead of runtimes.

Q8) what is SubGit?

SubGit is a tool for migrating SVN to Git. It creates a writable Git mirror of a local or remote Subversion repository and uses both Subversion and Git if you like.

Q9) how can you clone a Git repository via Jenkins?

First, we must enter the e-mail and user name for your Jenkins system, then switch into your job directory and execute the “git config” command.

Q10)What are the Advantages of Ansible?

Agentless, it doesn’t require any extra package/daemons to be installed

Very low overhead

Good performance

Idempotent

Very Easy to learn

Declarative not procedural

Q11) what’s the use of Ansible?

Ansible is mainly used in IT infrastructure to manage or deploy applications to remote nodes. Let’s say we want to deploy one application in 100’s of nodes by just executing one command, then Ansible is the one actually coming into the picture but should have some knowledge on Ansible script to understand or execute the same.

Q12) what's the difference between Ansible Playbook and Roles?

Roles	Playbooks
Roles are reusable subsets of a play.	Playbooks contain Plays.
A set of tasks for accomplishing certain role.	Mapps among hosts and roles.
Example: common, webservers.	Example: site.yml, fooservers.yml, webservers.yml.

Q13) How do I see a list of all the ansible_ variables?

Ansible by default gathers "facts" about the machines, and these facts can be accessed in Playbooks and in templates. To see a list of all the facts that are available about a machine, you can run the "setup" module as an ad-hoc action:

Ansible -m setup hostname

This will print out a dictionary of all the facts that are available for that particular host.

Q14) what is Docker?

Docker is a containerization technology that packages your application and all its dependencies together in the form of Containers to ensure that your application works seamlessly in any environment.

Q15) what is Docker image?

Docker image is the source of Docker container. Or in other words, Docker images are used to create containers.

Q16) what is Docker Container?

Docker Container is the running instance of Docker Image.

Q17) Can we consider DevOps as Agile methodology?

Of Course, we can!! The only difference between agile methodology and DevOps is that, agile methodology is implemented only for development section and DevOps implements agility on both development as well as operations section.

Q18) what are the advantages of using Git?

Data redundancy and replication

High availability

Only one. git directory per repository

Superior disk utilization and network performance

Collaboration friendly

Git can use any sort of projects.

Q19) what is kernel?

A kernel is the lowest level of easily replaceable software that interfaces with the hardware in your computer.

Q20) what is difference between grep -i and grep -v?

I ignore alphabet difference V accept this value

ex) ls | grep -i docker

Dockerfile

docker.tar.gz

ls | grep -v docker

Desktop

Dockerfile

Documents

Downloads

You can’t see anything with name docker.tar.gz

Q21) How can you define particular space to the file

This feature is generally used to give the swap space to the server. Lets say in below machine I have to create swap space of 1GB then,

dd if=/dev/zero of=/swapfile1 bs=1G count=1

Q22) what is concept of sudo in linux?

Sudo(superuser do) is a utility for UNIX- and Linux-based systems that provides an efficient way to give specific users permission to use specific system commands at the root (most powerful) level of the system.

Q23) what is a Jenkins Pipeline?

Jenkins Pipeline (or simply “Pipeline”) is a suite of plugins which supports implementing and integrating continuous delivery pipelines into Jenkins.

Q24) How to stop and restart the Docker container?

To stop the container: docker stop container ID

Now to restart the Docker container: docker restart container ID

Q25) What platforms does Docker run on?

Docker runs on only Linux and Cloud platforms:

Ubuntu 12.04 LTS+

Fedora 20+

RHEL 6.5+

CentOS 6+

Gentoo

ArchLinux

openSUSE 12.3+

CRUX 3.0+

Cloud:

Amazon EC2

Google Compute Engine

Microsoft Azure

Rackspace

Note that Docker does not run on Windows or Mac for production as there is no support, yes you can use it for testing purpose even in windows

Q26) what are the tools used for docker networking?

For docker networking we generally use kubernets and docker swarm.

Q27) what is docker compose?

Lets say you want to run multiple docker container, at that time you have to create the docker compose file and type the command docker-compose up. It will run all the containers mentioned in docker compose file.

Q28) What is Scrum?

Scrum is basically used to divide your complex software and product development task into smaller chunks, using iterations and incremental practises. Each iteration is of two weeks. Scrum consists of three roles: Product owner, scrum master and Team

Q29) What does the commit object contain?

Commit object contain the following components:

It contains a set of files, representing the state of a project at a given point of time reference to parent commit objects

An SHAI name, a 40-character string that uniquely identifies the commit object (also called as hash).

Q30) Explain the difference between git pull and git fetch?

Git pull command basically pulls any new changes or commits from a branch from your central repository and updates your target branch in your local repository.

Git fetch is also used for the same purpose, but its slightly different form Git pull. When you trigger a git fetch, it pulls all new commits from the desired branch and stores it in a new branch in your local repository. If we want to reflect these changes in your target branch, git fetch must be followed with a git merge. Our target branch will only be updated after merging the target branch and fetched branch. Just to make it easy for us, remember the equation below:

Git pull = git fetch + git merge

Q31) How do we know in Git if a branch has already been merged into master?

git branch –merged

The above command lists the branches that have been merged into the current branch.

git branch –no-merged

this command lists the branches that have not been merged.

Q32) What is ‘Staging Area’ or ‘Index’ in GIT?

Before committing a file, it must be formatted and reviewed in an intermediate area known as ‘Staging Area’ or ‘Indexing Area’.

#git add <file_name>

Q33) What is Git Stash?

Let’s say you’ve been working on part of your project, things are in a messy state and you want to switch branches for some time to work on something else. The problem is, you don’t want to do a commit of your half-done work just, so you can get back to this point later. The answer to this issue is Git stash.

Git Stashing takes your working directory that is, your modified tracked files and staged changes and saves it on a stack of unfinished changes that you can reapply at any time.

Q34) What is Git stash drop?

Git ‘stash drop’ command is basically used to remove the stashed item. It will basically remove the last added stash item by default, and it can also remove a specific item if you include it as an argument.

I have provided an example below:

If you want to remove any particular stash item from the list of stashed items you can use the below commands:

git stash list: It will display the list of stashed items as follows:

stash@{0}: WIP on master: 049d080 added the index file

stash@{1}: WIP on master: c265351 Revert “added files”

stash@{2}: WIP on master: 13d80a5 added number to log

Q35) What is the function of ‘git config’?

Git uses our username to associate commits with an identity. The git config command can be used to change our Git configuration, including your username.

Suppose you want to give a username and email id to associate commit with an identity so that you can know who has made a commit. For that I will use:

git config –global user.name “Your Name”: This command will add your username.

git config –global user.email “Your E-mail Address”: This command will add your email id.

Q36) How can you create a repository in Git?

To create a repository, you must create a directory for the project if it does not exist, then run command “git init”. By running this command .git directory will be created inside the project directory.

Q37) Describe the branching strategies you have used?

Generally, they ask this question to understand your branching knowledge

Feature branching

This model keeps all the changes for a feature inside of a branch. When the feature branch is fully tested and validated by automated tests, the branch is then merged into master.

Task branching

In this task branching model each task is implemented on its own branch with the task key included in the branch name. It is quite easy to see which code implements which task, just look for the task key in the branch name.

Release branching

Once the develop branch has acquired enough features for a release, then we can clone that branch to form a Release branch. Creating this release branch starts the next release cycle, so no new features can be added after this point, only bug fixes, documentation generation, and other release-oriented tasks should go in this branch. Once it’s ready to ship, the release gets merged into master and then tagged with a version number. In addition, it should be merged back into develop branch, which may have progressed since the release was initiated earlier.

Q38) What is Jenkins?

Jenkins is an open source continuous integration tool which is written in Java language. It keeps a track on version control system and to initiate and monitor a build system if any changes occur. It monitors the whole process and provides reports and notifications to alert the concern team.

Q39) What is the difference between Maven, Ant and Jenkins?

Maven and Ant are Build Technologies whereas Jenkins is a continuous integration(CI/CD) tool.

Q40) Explain what is continuous integration?

When multiple developers or teams are working on different segments of same web application, we need to perform integration test by integrating all the modules. To do that an automated process for each piece of code is performed on daily bases so that all your code gets tested. And this whole process is termed as continuous integration.

Q41) What is the relation between Hudson and Jenkins?

Hudson was the earlier name of current Jenkins. After some issue faced, the project name was changed from Hudson to Jenkins.

Q42) What are the advantages of Jenkins?

Advantage of using Jenkins

Bug tracking is easy at early stage in development environment.

Provides a very large numbers of plugin support.

Iterative improvement to the code, code is basically divided into small sprints.

Build failures are cached at integration stage.

For each code commit changes an automatic build report notification get generated.

To notify developers about build report success or failure, it can be integrated with LDAP mail server.

Achieves continuous integration agile development and test-driven development environment.

With simple steps, maven release project can also be automated.

Q43) Which SCM tools does Jenkins supports?

Source code management tools supported by Jenkins are below:

AccuRev

CVS

Subversion

Git

Mercurial

Perforce

Clearcase

RTC

Q44) What is Ansible?

Ansible is a software configuration management tool to deploy an application using ssh without any downtime. It is also used for management and configuration of software applications. Ansible is developed in Python language.

Q45) How can your setup Jenkins jobs?

Steps to set up Jenkins job as follows:

Select new item from the menu.

After that enter a name for the job (it can be anything) and select free-style job.

Then click OK to create new job in Jenkins dashboard.

The next page enables you to configure your job, and it’s done.

Q46) What is your daily activities in your current role?

Working on JIRA Tickets

Builds and Deployments

Resolving issues when builds and deployments fails by coordinating and collaborating with the dev team

Infrastructure maintenance

Monitoring health of applications

Q47) What are the challenges you faced in recent times?

I need to implement trending technologies like Docker to automate the configuration management activities in my project by showing POC.

Q48) What are the build and deployment failures you got and how you resolved those?

I use to get most of the time out of memory issue. So I fixed this issue by restarting the server which is not best practice. I did the permanent fix by increase the Perm Gen Space and Heap Space.

Q49) I want a file that consists of last 10 lines of the some other file?

Tail -10 filename >filename

Q50) How to check the exit status of the commands?

echo $?

Q51) I want to get the information from file which consists of the word “GangBoard”

grep “GangBoard” filename

Q52) I want to search the files with the name of “GangBoard”

find / -type f -name “*GangBoard*”

Q53) Write a shell script to print only prime numbers?

prime.sh

```
echo "1"
i=3
j=300
flag=0
tem=2
echo "1"while [ $i -ne $j ]
do
        temp=`echo $i`
        while [ $temp -ne $tem ]
        do
                temp=`expr $temp - 1`
                n=`expr $i % $temp`
                if [ $n -eq 0 -a $flag -eq 0 ]
                then
                        flag=1
                fi
        done
        if [ $flag -eq 0 ]
        then
```

```
        echo $i
    else
        flag=0
    fi
    i=`expr $i + 1`
done
```

Q54) How to pass the parameters to the script and how can I get those parameters?

Scriptname.sh parameter1 parameter2

I will use $* to get the parameters.

Q55) What is the default file permissions for the file and how can I modify it?

Default file permissions are : rw-r—r—

If I want to change the default file permissions I need to use umask command ex: umask 666

Q56) How you will do the releases?

There are some steps to follow.

Create a check list

Create a release branch

Bump the version

Merge release branch to master & tag it.

Use a Pull request to merge the release merge

Deploy master to Prod Environment

Merge back into develop & delete release branch

Change log generation

Communicating with stack holders

Grooming the issue tracker

Q57) How you automate the whole build and release process?

Check out a set of source code files.

Compile the code and report on progress along the way.

Run automated unit tests against successful compiles.

Create an installer.

Publish the installer to a download site, and notify teams that the installer is available.

Run the installer to create an installed executable.

Run automated tests against the executable.

Report the results of the tests.

Launch a subordinate project to update standard libraries.

Promote executables and other files to QA for further testing.

Deploy finished releases to production environments, such as Web servers or CD manufacturing.

The above process will be done by Jenkins by creating the jobs.

Q58) I have 50 jobs in the Jenkins dash board , I want to build at a time all the jobs

In Jenkins there is a plugin called build after other projects build. We can provide job names over there and If one parent job run then it will automatically run the all other jobs. Or we can use Pipe line jobs.

Q59) How can I integrate all the tools with Jenkins?

I have to navigate to the manage Jenkins and then global tool configurations there you have to provide all the details such as Git URL , Java version, Maven version , Path etc.

Q60) How to install Jenkins via Docker?

The steps are:

Open up a terminal window.

Download the jenkinsci/blueocean image & run it as a container in Docker using the following docker run command:(https://docs.docker.com/engine/reference/commandline/run/)

docker run \ -u root \ –rm \ -d \ -p 8080:8080 \ -p 50000:50000 \ -v jenkins-data:/var/jenkins_home \ -v /var/run/docker.sock:/var/run/docker.sock \ jenkinsci/blueocean

Proceed to the Post-installation setup wizard (https://jenkins.io/doc/book/installing/#setup-wizard)

Accessing the Jenkins/Blue Ocean Docker container docker exec -it jenkins-blueocean bash

Accessing the Jenkins console log through Docker logsdocker logs <docker-container-name>Accessing the Jenkins home directorydocker exec -it <docker-container-name> bash

Q61) Did you ever participated in Prod Deployments? If yes what is the procedure?

Yes I have participated, we need to follow the following steps in my point of view

Preparation & Planning : What kind of system/technology was supposed to run on what kind of machine

The specifications regarding the clustering of systems

How all these stand-alone boxes were going to talk to each other in a foolproof manner

Production setup should be documented to bits. It needs to be neat, foolproof, and understandable.

It should have all a system configurations, IP addresses, system specifications, & installation instructions.

It needs to be updated as & when any change is made to the production environment of the system

Q62) My application is not coming up for some reason? How can you bring it up?

We need to follow the steps

Network connection

The Web Server is not receiving users’s request

Checking the logs

Checking the process id’s whether services are running or not

The Application Server is not receiving user’s request(Check the Application Server Logs and Processes)

A network level ‘connection reset’ is happening somewhere.

Q63) Did you automate anything in your project? Please explain

Yes I have automated couple of things such as

Password expiry automation

Deleting the older log files

Code quality threshold violations etc.

Q64) What is IaC? How you will achieve this?

Infrastructure as Code (IaC) is the management of infrastructure (networks, virtual machines, load balancers, and connection topology) in a descriptive model, using the same versioning as DevOps team uses for source code. This will be achieved by using the tools such as Chef, Puppet and Ansible etc.

Q65) What is multifactor authentication? What is the use of it?

Multifactor authentication (MFA) is a security system that requires more than one method of authentication from independent categories of credentials to verify the user's identity for a login or other transaction.

Security for every enterprise user — end & privileged users, internal and external

Protect across enterprise resources — cloud & on-prem apps, VPNs, endpoints, servers, privilege elevation and more

Reduce cost & complexity with an integrated identity platform

Q66) I want to copy the artifacts from one location to another location in cloud. How?

Create two S3 buckets, one to use as the source, and the other to use as the destination and then create policies.

Q67) How can I modify the commit message in git?

I have to use following command and enter the required message.

Git commit –amend

Q68) How can you avoid the waiting time for the triggered jobs in Jenkins.

First I will check the Slave nodes capacity, If it is fully loaded then I will add the slave node by doing the following process.

Go to the Jenkins dashboard -> Manage Jenkins ->Manage Nodes

Create the new node a

By giving the all required fields and launch the slave machine as you want.

Q69) What are the Pros and Cons of Ansible?

Pros:

Open Source

Agent less

Improved efficiency , reduce cost

Less Maintenance

Easy to understand yaml files

Cons:

Underdeveloped GUI with limited features

Increased focus on orchestration over configuration management

SSH communication slows down in scaled environments

Q70) How you handle the merge conflicts in git?

Follow the steps

Create Pull request

Modify according to the requirement by sitting with developers

Commit the correct file to the branch

Merge the current branch with master branch.

Q71) I want to delete 10 days older log files. How can I?

There is a command in unix to achieve this task find <directory_path> -mtime +10 -name “*.log” -exec rm -f {} \; 2>/dev/null

What is the difference among chef, puppet and ansible?

	Chef	Puppet	Ansible
Interoperability	Works Only on Linux/Unix	Works Only on Linux/Unix	Supports Windows but server should be Linux/U
Conf. Language	It uses Ruby	Puppet DSL	YAML (Python)
Availability	Primary Server and Backup Server	Multi Master Architecture	Single Active Node

Q72) How you get the Inventory variables defined for the host?

We need to use the following command

Ansible – m debug- a “var=hostvars[‘hostname’]” localhost(10.92.62.215)

Q73) How you will take backup for Jenkins?

Copy JENKINS_HOME directory and “jobs” directory to replicate it in another server

Q74) How to deploy docker container to aws?

Amazon provides the service called Amazon Elastic Container Service; By using this creating and configuring the task definition and services we will launch the applications.

Q75) I want to change the default port number of apache tomcat. How?

Go to the tomcat folder and navigate to the conf folder there you will find a server.xml file. You can change connector port tag as you want.

Q76) In how many ways you can install the Jenkins?

We can install Jenkins in 3 Ways

By downloading Jenkins archive file

By running as a service Java –jar Jenkins.war

By deploying Jenkins.war to the webapps folder in tomcat.

Q77) How you will run Jenkins job from command line?

We have a Jenkins CLI from there we need to use the curl command

curl -X POST -u YOUR_USER:YOUR_USER_PASSWORD http://YOUR_JENKINS_URL/job/YOUR_JOB/build

Q78) How you will do tagging in git?

We have following command to create tags in git

Git tag v0.1

Q79) How can you connect a container to a network when it starts?

We need to use a following command

docker run -itd –network=multi-host-network busybox

Q80) How you will do code commit and code deploy in cloud?

Create a deployment environment

Get a copy of the sample code

Create your pipeline

Activate your pipeline

Commit a change and update the App.

Q81) How to access variable names in Ansible?

Using hostvars method we can access and add the variables like below

{{ hostvars[inventory_hostname][‘ansible_’ + which_interface][‘ipv4’][‘address’] }}

Q82) What is Infrastructure as Code?

Where the Configuration of any servers or tool chain or application stack required for an association can be made into progressively elucidating dimension of code and that can be utilized for provisioning and overseeing foundation components like Virtual Machine, Software, Network Elements, however it varies from contents utilizing any language, where they are a progression of static advances coded, where Version control can be utilized so as to follow condition changes .

Precedent Tools are Ansible, Terraform.

Q83) What are the zones the Version control can acquaint with get proficient DevOps practice?

A clearly fundamental region of Version Control is Source code the executives, Where each engineer code ought to be pushed to a typical storehouse for keeping up assemble and discharge in CI/CD pipelines.

Another territory can be Version control For Administrators when they use Infrastructure as A Code (IAC) apparatuses and rehearses for keeping up The Environment setup.

Another Area of Version Control framework Can be Artifactory Management Using Repositories like Nexus and DockerHub

Q84) Why Opensource apparatuses support DevOps?

Opensource devices dominatingly utilized by any association which is adjusting (or) embraced DevOps pipelines in light of the fact that devops accompanied an attention on robotization in different parts of association manufacture and discharge and change the executives and furthermore framework the board zones.

So creating or utilizing a solitary apparatus is unthinkable and furthermore everything is fundamentally an experimentation period of advancement and furthermore coordinated chops down the advantage of building up a solitary device , so opensource devices were accessible available practically spares each reason and furthermore gives association a choice to assess the device dependent on their need.

Q85) What is the distinction among Ansible and chef(or) manikin?

Ansible is Agentless design the board device, where manikin or gourmet expert needs operator should be kept running on the specialist hub and culinary specialist or manikin depends on draw demonstrate, where your cookbook or show for gourmet expert and manikin separately from the ace will be pulled by the operator and ansible uses ssh to convey and it gives information driven guidelines to the hubs should be overseen , progressively like RPC execution, ansible utilizations YAML scripting, though manikin (or) culinary specialist is worked by ruby uses their own DSL .

Q86) What is Jinja2 templating in ansible playbooks and their utilization?

Jinja2 templating is the Python standard for templating , consider it like a sed editorial manager for Ansible , where it very well may be utilized is when there is a requirement for dynamic change of any config record to any application like consider mapping a MySQL application to the IP address of the machine, where it is running, it can’t be static , it needs modifying it progressively at runtime.

Arrangement

The vars inside the supports are supplanted by ansible while running utilizing layout module.

Q87) What is the requirement for sorting out playbooks as the job, is it vital?

Arranging playbooks as jobs , gives greater clarity and reusability to any plays , while consider an errand where MySQL establishment ought to be done after the evacuation of Oracle DB , and another prerequisite is expected to introduce MySQL after java establishment, in the two cases we have to introduce MySQL , yet without jobs need to compose playbooks independently for both use cases , yet utilizing jobs once the MySQL establishment job is made can be used any number of times by summoning utilizing rationale in site.yaml .

No, it isn't important to make jobs for each situation, however making jobs is the best practice in Ansible.

Q88) What is the fundamental disservice of Docker holders?

As the lifetime of any compartments is while pursuing a holder is wrecked you can't recover any information inside a compartment, the information inside a compartment is lost perpetually, however tenacious capacity for information inside compartments should be possible utilizing volumes mount to an outer source like host machine and any NFS drivers.

Q89) What are the docker motor and docker form?

Docker motor contacts the docker daemon inside the machine and makes the runtime condition and procedure for any compartment, docker make connects a few holders to shape as a stack utilized in making application stacks like LAMP, WAMP, XAMP

Q90) What are the Different modes does a holder can be run?

Docker holder can be kept running in two modes

Connected: Where it will be kept running in the forefront of the framework you are running, gives a terminal inside to compartment when – t choice is utilized with it, where each log will be diverted to stdout screen.

Isolates: This mode is typically kept running underway, where the holder is confined as a foundation procedure and each yield inside a compartment will be diverted log records inside/var/lib/docker/logs/<container-id>/<container-id.json> and which can be seen by docker logs order.

Q91) What the yield of docker assess order will be?

Docker examines <container-id> will give yield in JSON position, which contains subtleties like the IP address of the compartment inside the docker virtual scaffold and volume mount data and each other data identified with host (or) holder explicit like the basic document driver utilized, log driver utilized.

docker investigate [OPTIONS] NAME|ID [NAME|ID…]

Choices

Name, shorthand Default Description

group, – f Format the yield utilizing the given Go layout

measure, – s Display all out document sizes if the sort is the compartment

type Return JSON for a predefined type

Q92) What is the order can be utilized to check the asset usage by docker holders?

Docker details order can be utilized to check the asset usage of any docker holder, it gives the yield practically equivalent to Top direction in Linux, it shapes the base for compartment asset observing instruments like a counsel, which gets yield from docker details order.

docker details [OPTIONS] [CONTAINER…]

Choices

Name, shorthand Default Description

all, – a Show all holders (default demonstrates simply running)

group Pretty-print pictures utilizing a Go layout

no-stream Disable spilling details and just draw the main outcome

no-trunc Do not truncate yield

Q93) How to execute some errand (or) play on localhost just while executing playbooks on various has on an ansible?

In ansible, there is a module called delegate_to, in this module area give the specific host (or) has where your errands (or) assignment should be run.

undertakings:

name: ” Elasticsearch Hitting”

uri: url=’_search?q=status:new’ headers='{“Content-type”:”application/json”}’ method=GET return_content=yes

register: yield

delegate_to: 127.0.0.1

Q94) What is the distinction among set_fact and vars in ansible?

Where a set_fact sets the incentive for a factor at one time and stays static, despite the fact that the esteem is very powerful and vars continue changing according to the esteem continues changing for the variable.

assignments:

set_fact:

fact_time: “Truth: ”

troubleshoot: var=fact_time

order: rest 2

troubleshoot: var=fact_time

assignments:

name: queries in factors versus queries in realities

has: localhost

vars:

var_time: “Var: ”

Despite the fact that the query for the date has been utilized in both the cases, wherein the vars are utilized it modifies dependent on an opportunity to time each time executed inside the playbook lifetime. Be that as it may, Fact dependably continues as before once query is finished

Q95) What is a query in ansible and what are query modules bolstered by ansible?

Query modules enable access to information in Ansible from outside sources. These modules are assessed on the Ansible control machine and can incorporate perusing the filesystem yet in addition reaching outside information stores and administrations.

Organization is {lookup{‘<plugin>’,'<source(or)connection_string>’}}

A portion of the query modules upheld by ansible are

Document

pipe

redis

jinja layouts

etcd kv store

Q96) How might you erase the docker pictures put away at your nearby machine and how might you do it for every one of the pictures without a moment’s delay?

The direction docker RMI <image-id> can be utilized to erase the docker picture from nearby machine, though a few pictures may should be constrained in light of the fact that the picture might be utilized by some other holder (or) another picture , to erase pictures you can utilize the mix of directions by docker RMI $(docker pictures – q), where docker pictures will give the docker picture names, to get just the ID of docker pictures just , we are utilizing – q switch with docker pictures order.

Q97) What are the organizers in the Jenkins establishment and their employments?

JENKINS_HOME – which will be/$JENKINS_USER/.jenkins it is the root envelope of any Jenkins establishment and it contains subfolders each for various purposes.

employments/ – Folder contains all the data pretty much every one of the occupations arranged in the Jenkins example.

Inside employments/, you will have the envelope made for each activity and inside those organizers, you will have fabricate organizers as indicated by each form numbers each form will have its log records, which we see in Jenkins web support.

Modules/ – where all your modules will be recorded.

Workspace/ – this will be available to hold all the workspace documents like your source code pulled from SCM.

Q98) What are the approaches to design Jenkins framework?

Jenkins can be designed in two different ways

Web: Where there is a choice called design a framework, in their area, you can make all setup changes.

Manual on filesystem: Where each change should likewise be possible straightforwardly on the Jenkins config.xml document under the Jenkins establishment catalog, after you make changes on the filesystem, you have to restart your Jenkins, either can do it specifically from terminal (or) you can utilize Reload setup from plate under oversee Jenkins menu or you can hit/restart endpoint straightforwardly.

Q99) What is the job Of HTTP REST API in DevOps?

As DevOps is absolutely centers around Automating your framework and gives changes over the pipeline to various stages like an every CI/CD pipeline will have stages like form, test, mental soundness test, UAT, Deployment to Prod condition similarly as with each phase there are diverse devices is utilized and distinctive innovation stack is displayed and there should be an approach to incorporate with various instrument for finishing an arrangement toolchain, there comes a requirement for HTTP API , where each apparatus speaks with various devices utilizing API , and even client can likewise utilize SDK to interface with various devices like BOTOX for Python to contact AWS API’s for robotization dependent on occasions , these days its not cluster handling any longer , it is generally occasion driven pipelines

Q100) What are Micro services, and how they control proficient DevOps rehearses?

Where In conventional engineering , each application is stone monument application implies that anything is created by a gathering of designers, where it has been sent as a solitary application in numerous machines and presented to external world utilizing load balances, where the micro services implies separating your application into little pieces, where each piece serves the distinctive capacities expected to finish a solitary exchange and by separating , designers can likewise be shaped to gatherings and each bit of utilization may pursue diverse rules for proficient advancement stage, as a result of spry improvement ought to be staged up a bit and each administration utilizes REST API (or) Message lines to convey between another administration.

So manufacture and arrival of a non-strong form may not influence entire design, rather, some usefulness is lost, that gives the confirmation to productive and quicker CI/CD pipelines and DevOps Practices.

Q101) What are the manners in which that a pipeline can be made in Jenkins?

There are two different ways of a pipeline can be made in Jenkins

Scripted Pipelines:

Progressively like a programming approach

Explanatory pipelines:

DSL approach explicitly to make Jenkins pipelines.

The pipeline ought to be made in Jenkins document and the area can either be in SCM or nearby framework.

Definitive and Scripted Pipelines are developed on a very basic level in an unexpected way. Definitive Pipeline is a later element of Jenkins Pipeline which:

gives more extravagant grammatical highlights over Scripted Pipeline language structure, and is intended to make composing and perusing Pipeline code less demanding.

Q102) What are the Labels in Jenkins and where it tends to be used?

Similarly as with CI/CD arrangement should be concentrated , where each application in the association can be worked by a solitary CI/CD server , so in association there might be various types of utilization like java, c#,.NET and so forth, likewise with microservices approach your programming stack is inexactly coupled for the task , so you can have Labeled in every hub and select the choice Only assembled employments while name coordinating this hub, so when a manufacture is planned with the mark of the hub present in it, it hangs tight for next agent in that hub to be accessible, despite the fact that there are different agents in hubs.

Q103) What is the utilization of Blueocean in Jenkins?

Blue Ocean reconsiders the client experience of Jenkins. Planned from the beginning for Jenkins Pipeline, yet at the same time good with free-form occupations, Blue Ocean diminishes mess and builds lucidity for each individual from the group.

It gives complex UI to recognize each phase of the pipeline and better pinpointing for issues and extremely rich Pipeline editorial manager for apprentices.

Q104) What is the callback modules in Ansible, give a few instances of some callback modules?

Callback modules empower adding new practices to Ansible when reacting to occasions. Of course, callback modules control a large portion of the yield you see when running the direction line programs, however can likewise be utilized to include an extra yield, coordinate with different apparatuses and marshall the occasions to a capacity backend. So at whatever point a play is executed and after it creates a few occasions, that occasions are imprinted onto Stdout screen, so callback module can be put into any capacity backend for log preparing.

Model callback modules are ansible-logstash, where each playbook execution is brought by logstash in the JSON group and can be incorporated some other backend source like elasticsearch.

Q105) What are the scripting dialects can be utilized in DevOps?

As with scripting dialects, the fundamental shell scripting is utilized to construct ventures in Jenkins pipelines and python contents can be utilized with some other devices like Ansible , terraform as a wrapper content for some other complex choice unraveling undertakings in any mechanization as python is more unrivaled in complex rationale deduction than shell contents and ruby contents can likewise be utilized as fabricate ventures in Jenkins.

Q106) What is Continuous Monitoring and why checking is basic in DevOps?

DevOps draws out each association capacity of fabricate and discharge cycle to be a lot shorter with an idea of CI/CD, where each change is reflected into generation conditions fastly, so it should be firmly observed to get client input. So the idea of constant checking has been utilized to assess every application execution progressively (at any rate Near Real Time) , where every application is produced with application execution screen specialists perfect and the granular dimension of measurements are taken out like JVM details and even practical savvy measurements inside the application can likewise be spilled out progressively to Agents , which thusly provides for any backend stockpiling and that can be utilized by observing groups in dashboards and cautions to get persistently screen the application.

Q107) Give a few instances of persistent observing instruments?

Where numerous persistent observing instruments are accessible in the market, where utilized for an alternate sort of use and sending model

Docker compartments can be checked by consultant operator, which can be utilized by Elasticsearch to store measurements (or) you can utilize TICK stack (Telegraph, influxdb, Chronograph, Capacitor) for each framework observing in NRT(Near Real Time) and You can utilize Logstash (or) Beats to gather Logs from framework , which thusly can utilize Elasticsearch as Storage Backend can utilize Kibana (or) Grafana as visualizer.

The framework observing should be possible by Nagios and Icinga.

Q108) What is docker swarm?

Gathering of Virtual machines with Docker Engine can be grouped and kept up as a solitary framework and the assets likewise being shared by the compartments and docker swarm ace calendars the docker holder in any of the machines under the bunch as indicated by asset accessibility

Docker swarm init can be utilized to start docker swarm bunch and docker swarm joins with the ace IP from customer joins the hub into the swarm group.

Q109) What are Microservices, and how they control productive DevOps rehearses?

Where In conventional engineering , each application is stone monument application implies that anything is created by a gathering of designers, where it has been conveyed as a solitary application in numerous machines and presented to external world utilizing load balancers, where the microservices implies separating your application into little pieces, where each piece serves the diverse capacities expected to finish a solitary exchange and by separating , engineers can likewise be shaped to gatherings and each bit of utilization may pursue distinctive rules for proficient advancement stage, on account of light-footed improvement ought to be staged up a bit and each administration utilizes REST API (or) Message lines to impart between another administration.

So manufacture and arrival of a non-hearty variant may not influence entire design, rather, some usefulness is lost, that gives the affirmation to proficient and quicker CI/CD pipelines and DevOps Practices.

Q110) What are the manners in which that a pipeline can be made in Jenkins?

There are two different ways of a pipeline can be made in Jenkins

Scripted Pipelines:

Progressively like a programming approach

Explanatory pipelines:

DSL approach explicitly to make Jenkins pipelines.

The pipeline ought to be made in Jenkins record and the area can either be in SCM or neighborhood framework.

Definitive and Scripted Pipelines are developed in a general sense in an unexpected way. Explanatory Pipeline is a later element of Jenkins Pipeline which:

gives more extravagant linguistic highlights over Scripted Pipeline sentence structure, and is intended to make composing and perusing Pipeline code simpler.

Q111) What are the Labels in Jenkins and where it very well may be used?

Likewise with CI/CD arrangement should be incorporated , where each application in the association can be worked by a solitary CI/CD server , so in association there might be various types of use like java, c#,.NET and so forth, similarly as with microservices approach your programming stack is inexactly coupled for the undertaking , so you can have Labeled in every hub and select the alternative Only assembled occupations while mark coordinating this hub, so when a fabricate is booked with the name of the hub present in it, it sits tight for next agent in that hub to be accessible, despite the fact that there are different agents in hubs.

Q112) What is the utilization of Blueocean in Jenkins?

Blue Ocean reexamines the client experience of Jenkins. Planned starting from the earliest stage for Jenkins Pipeline, yet at the same time good with free-form occupations, Blue Ocean lessens mess and expands clearness for each individual from the group.

It gives modern UI to recognize each phase of the pipeline and better pinpointing for issues and rich Pipeline proofreader for fledglings.

Q113) What is the callback modules in ansible, give a few instances of some callback modules?

Callback modules empower adding new practices to Ansible when reacting to occasions. As a matter of course, callback modules control the greater part of the yield you see when running the direction line programs, yet can likewise be utilized to include an extra yield, coordinate with different instruments and marshall the occasions to a capacity backend. So at whatever point a play is executed and after it delivers a few occasions, that occasions are imprinted onto Stdout screen, so callback module can be put into any capacity backend for log handling.

Precedent callback modules are ansible-logstash, where each playbook execution is gotten by logstash in the JSON position and can be incorporated some other backend source like elasticsearch.

Q114) What are the scripting dialects can be utilized in DevOps?

As with scripting dialects, the fundamental shell scripting is utilized to assemble ventures in Jenkins pipelines and python contents can be utilized with some other instruments like Ansible.

Q115) For what reason is each instrument in DevOps is generally has some DSL (Domain Specific Language)?

Devops is a culture created to address the necessities of lithe procedure, where the advancement rate is quicker ,so sending should coordinate its speed and that needs activities group to arrange and work with dev group, where everything can computerize utilizing content based , however it feels more like tasks group than , it gives chaotic association of any pipelines, more the utilization cases , more the contents should be composed , so there are a few use cases, which will be sufficient to cover the requirements of light-footed are taken and apparatuses are made by that and customization can occur over the device utilizing DSL to mechanize the DevOps practice and Infra the board.

Q116) What are the mists can be incorporated with Jenkins and what are the utilization cases?

Jenkins can be coordinated with various cloud suppliers for various use cases like dynamic Jenkins slaves, Deploy to cloud conditions.

A portion of the cloud can be incorporated are

AWS

Purplish blue

Google Cloud

OpenStack

Q117) What are Docker volumes and what sort of volume ought to be utilized to accomplish relentless capacity?

Docker volumes are the filesystem mount focuses made by client for a compartment or a volume can be utilized by numerous holders, and there are distinctive sorts of volume mount accessible void dir, Post mount, AWS upheld lbs volume, Azure volume, Google Cloud (or) even NFS, CIFS filesystems, so a volume ought to be mounted to any of the outer drives to accomplish determined capacity, in light of the fact that a lifetime of records inside compartment, is as yet the holder is available and if holder is erased, the information would be lost.

Q118) What are the Artifacts store can be incorporated with Jenkins?

Any sort of Artifacts vault can be coordinated with Jenkins, utilizing either shell directions (or) devoted modules, some of them are Nexus, Jfrog.

Q119) What are a portion of the testing apparatuses that can be coordinated with Jenkins and notice their modules?

Sonar module – can be utilized to incorporate testing of Code quality in your source code.

Execution module – this can be utilized to incorporate JMeter execution testing.

Junit – to distribute unit test reports.

Selenium module – can be utilized to incorporate with selenium for computerization testing.

Q120) What are the manufacture triggers accessible in Jenkins?

Fabricates can be run physically (or) either can naturally be activated by various sources like

Webhooks- The webhooks are API calls from SCM, at whatever point a code is submitted into a vault (or) should be possible for explicit occasions into explicit branches.

Gerrit code survey trigger-Gerrit is an opensource code audit instrument, at whatever point a code change is endorsed after audit construct can be activated.

Trigger Build Remotely – You can have remote contents in any machine (or) even AWS lambda capacities (or) make a post demand to trigger forms in Jenkins.

Calendar Jobs-Jobs can likewise be booked like Cron occupations.

Survey SCM for changes – Where your Jenkins searches for any progressions in SCM for the given interim, if there is a change, a manufacture can be activated.

Upstream and Downstream Jobs-Where a construct can be activated by another activity that is executed already.

Q121) How to Version control Docker pictures?

Docker pictures can be form controlled utilizing Tags, where you can relegate the tag to any picture utilizing docker tag <image-id> order. Furthermore, on the off chance that you are pushing any docker center library without labeling the default label would be doled out which is most recent, regardless of whether a picture with the most recent is available, it indicates that picture without the tag and reassign that to the most recent push picture.

Q122) What is the utilization of Timestamper module in Jenkins?

It adds Timestamp to each line to the comfort yield of the assemble.

Q123) Why you ought not execute an expand on ace?

You can run an expand on ace in Jenkins , yet it isn’t prudent, in light of the fact that the ace as of now has the duty of planning assembles and getting incorporate yields with JENKINS_HOME index, so on the off chance that we run an expand on Jenkins ace, at that point it furthermore needs to manufacture apparatuses, and workspace for source code, so it puts execution over-burden in the framework, if the Jenkins ace accidents, it expands the downtime of your fabricate and discharge cycle.

Q124) What do the main benefits of DevOps?

With a single team composed of cross-functional comments simply working in collaboration, DevOps organizations container produce including maximum speed, functionality, including innovation. Where continue special benefits: Continuous software control. Shorter complexity to manage.

Q125) What are the uses of DevOps tools?

Gradle. Your DevOps device stack will need a reliable build tool.

Git. Git is one from the most successful DevOps tools, widely applied across the specific software industry.

Jenkins. Jenkins is that go-to DevOps automation tool for many software community teams.

Bamboo.

Docker.

Kubernetes.

Puppet Enterprise.

Ansible.

Q126) What is DevOps beginner?

DevOps is a society which supports collaboration between Development including Operations Team to deploy key to increase faster in an automated & repeatable way. In innocent words, DevOps backside is established as an association of development and IT operations including excellent communication and collaboration.

Q127) What is the roles and responsibilities of the DevOps engineer?

DevOps Engineer manages with developers including the IT system to manage the code releases. They are both developers cases become interested in deployment including practice settings or sysadmins who convert a passion for scripting and coding more move toward the development front where all can improve that planning from test and deployment.

Q128) Which is the top DevOps tools? and it's Which tools have you worked on?

Discover about the trending Top DevOps Tools including Git. Well, if you live considering DevOps being a tool when, you are wrong! DevOps does not a tool or software, it's an appreciation that you can adopt for continuous growth. file and, by practicing it you can simply coordinate this work among your team.

Q129) Explain the typical characters involved in DevOps?

Commitment to the superior level in the organization.

Need for silver to be delivered across the organization.

Version check software.

Automated tools to compliance to process.

Automated Testing

Automated Deployment

Q130) What are your expectations from a career perspective of DevOps?

To be involved in the end to end delivery method and the most important phase of helping to change the manner so as to allow that development and operations teams to go together also understand each other's point of view.

Q131) What does configuration management under terms like infrastructure further review some popular tools used?

In Software Engineering Software Configuration Management is a unique task about tracking to make the setting configuration during the infrastructure with one change. It is done for deploying, configuring and maintaining servers.

Q132) How will you approach when each design must to implement DevOps?

As the application is generated and deployed, we do need to control its performance. Monitoring means also really important because it might further to uncover some defects which might not have been detected earlier.

Q133) Explain about from Continuous Testing

From the above goal of Continuous Integration which is to take this application excuse to close users are primarily providing continuous delivery. This backside is completed out any adequate number about unit testing and automation testing. Hence, we must validate that this system created and integrated with all the developers that work as required.

Q134) Explain about from Continuous Delivery.

Continuous Delivery means an extension of Constant Integration which primarily serves to make the features which some developers continue developing out on some end users because soon as possible. During this process, it passes through several stages of QA, Staging etc., and before for delivery to the PRODUCTION system.

Q135) What are the tasks also responsibilities of DevOps engineer?

In this role, you'll work collaboratively including software engineering to use and operate our systems. Help automate also streamline our procedures and processes. Build also maintain tools for deployment, monitoring, including operations. And troubleshoot and resolve problems in our dev, search and production environments.

Q136) What is defined DevOps engineer should know?

DevOps Engineer goes including developers and that IT staff to manage this code releases. They live both developers who become involved through deployment including web services or sysadmins that become a passion for scripting and coding more move into the development design where only can develop this planning from search also deployment.

Q137) How much makes any DevOps engineer make?

A lead DevOps engineer can get between $137,000 including $180,000, according to April 2018 job data of Glassdoor. The common salary from any lead DevOps engineer based at the Big Apple is $141,452.

Q138) What mean the specific skills required for a DevOps engineer?

While tech abilities are a must, strong DevOps engineers further possess this ability to collaborate, multi-task, also always place that customer first. critical skills that all DevOps engineer requirements for success.

Q139) What is DevOps also why is it important?

Implementing the new approach would take in many advantages on an organization. A seamless collection up can be performed in the teams of developers, test managers, and operational executives also hence they can work in collaboration including each other to achieve a greater output on a project.

Q140) What is means by DevOps lifecycle?

DevOps means an agile connection between development including operations. It means any process followed by this development because well because of help drivers clean of this starting of this design to production support. Understanding DevOps means incomplete excuse estimated DevOps lifecycle.

Tools for an efficient DevOps workflow. A daily workflow based at DevOps thoughts allows team members to achieve content faster, be flexible just to both experiments also deliver value, also help each part from this organization use a learning mentality.

Q142) Can you make DevOps without agile?

DevOps is one about some key elements to assist you to achieve this. Can you do agile software evolution without doing DevOps But managing agile software development and being agile are a couple really different things.

Q143) What exactly defined is DevOps?

DevOps is all of bringing commonly the structure also process of traditional operations, so being support deployment, including any tools, also practices of traditional construction methods so as source control also versioning.

Q144) Need for Continuous Integration:

Improves the quality of software.

Reduction in time taken to delivery

Allows dev team to detect and locate problems early

Q145) Success factor for the Continuous Integration

Maintain Code Repository

Automate the build

Perform daily checkin and commits to baseline

Test in clone environment

Keep the build fast

Make it easy to get the newest deliverables

Q146) Can we copy Jenkins job from one server to other server?

Yes, we can do that using one of the following ways

We can copy the Jenkins jobs from one server to other server by copying the corresponding jobs folder.

We can make a copy of the existing job by making clone of a job directory with different names

Rename the existing job by renaming the directory

Q147) How can we create the backup and copy in Jenkins?

We can copy or backup, we need to backup JENKINS_HOME directory which contains the details of all the job configurations, build details etc.

Q148) Difference between “poll scm” and “build periodically”

Poll SCM will trigger the build only if it detects the change in SCM, whereas Build Periodically will trigger the build once the given time period is elapsed.

Q149) What is difference between docker image and docker container?

Docker image is a readonly template that contains the instructions for a container to start.

Docker container is a runnable instance of a docker image

Q150) What is Application Containerization?

It is a process of OS Level virtualization technique used to deploy the application without launching the entire VM for each application where multiple isolated applications or services can access the same Host and run on the same OS.

Q151) syntax for building docker image

docker build –f <filename> -t imagename:version

Q152) running docker image

docker run –dt –restart=always –p <hostport>:<containerport> -h <hostname> -v <hostvolume>:<containervolume> imagename:version

Q153) How to log into a container

docker exec –it <containerID> /bin/bash

Q154) What is Puppet?

Puppet is a Configuration Management tool, Puppet is used to automate administration tasks.

Q155) What is Configuration Management?

Configuration Management is the System engineering process. Configuration Management applied over the life cycle of a system provides visibility and control of its performance, functional, and physical attributesrecording their status and in support of Change Management.

Q156) List the Software Configuration Management Features.

Enforcement

Cooperating Enablement

Version Control Friendly

Enable Change Control Processes

Q157) List out the 5 Best Software Configuration Management Tools.

CFEngine Configuration Tool.

CHEF Configuration Tool

Ansible Configuration Tool

Puppet Configuration Tool.

SALTSTACK Configuration Tool.

Q158) Why should Puppet be chosen?

It has good community support

Easy to Learn Programming Language DSL

It is open source

Q159) What is Saltstack?

SaltStack is based on Python programming & Scripiting language. Its also a configuration tool.Saltstack works on a non-centralized model or a master-client setup model. it provides a push and SSH methods to communicate with clients.

Q160) Why should Puppet to be chosen?

There are Some Reason puppet to be chosen.

Puppet is open source

Easy to Learn Programming Language DSL

Puppet has good community support

Q161) Advantages of VCS

Multiple people can work on the same project and it helps us to keep track of the files and documents and their changes.

We can merge the changes from multiple developers to single stream.

Helps us to revert to the earlier version if the current version is broke.

Helps us to maintain multiple version of the software at the same location without rewriting.

Q162) Advantages of DevOps

Below are the major advantages

Technical:

Continuous software delivery

Less Complexity

Faster Resolution

Business:

Faster delivery of the features

More stable operating environment

Improved communication and collaboration between various teams

Q163) Use cases where we can use DevOps

Explain the legacy / old procedures that are followed to develop and deploy software

Problems of that approach

How can we solve the above issues using DevOps.

For the 1st and 2nd points, development of the application, problems in build and deployment, problems in operations, problems in debugging and fixing the issues

For 3rd point explain various technologies we can use to ease the deployments, for development, explain about taking small features and development, how it helps for testing and issue fixing.

Q164) Major difference between Agile and DevOps

Agile is the set of rules/principles and guidelines about how to develop a software. There are chances that this developed software works only on developer's environment. But to release that software to public consumption and deploy in production environment, we will use the DevOps tools and Techniques for the operation of that software.

In a nutshell, Agile is the set of rules for the development of a software, but DevOps focus more on Development as well as Operation of the Developed software in various environments.

Q165) What Are the Benefits Of Nosql?

Non-relationals and schema-less data models

Low latency and high performance

Highly scalable

Q166) What Are Adoptions Of Devops In Industry?

Use of the agile and other development processes and methods.

Demand for an increased rate of the production releases from application and business.

Wide availability of virtuals and cloud infrastructure from both internal and external providers;

Increased usage of the data center ,automation and configuration management tools;

Increased focus on the test automation and continuous integration methods;

Best practices on the critical issues.

Q167) How Is the Chef Used As a Cm Tool ?

Chef is the considered to be one of the preferred industry-wide CM tools. Facebook migrated its an infrastructure and backend IT to the Chef platform, for example. Explain how to the Chef helps you to avoid delays by automating processes. The scripts are written in Ruby. It can integrate with a cloud-based platforms and configure new systems. It provides many libraries for the infrastructure development that can later to be deployed within a software. Thanks to its centralized management system, one of the Chef server is enough to be used as the center for deploying various policies.

Q168) Why Are the Configuration Management Processes And Tools Important ?

Talk about to multiple software builds, releases, revisions, and versions for each other software or testware that is being developed. Move on to explain the need for storing and maintaining data, keeping track of the development builds and simplified troubleshooting. Don't forget to mention that key CM tools that can be used to the achieve these objectives. Talk about how to tools like Puppet, Ansible, and Chef help in automating software deployment and configuration on several servers.

Q169) Which Are the Some Of the Most Popular Devops Tools ?

The most popular DevOps tools included`

Selenium

Puppet

Chef

Git

Jenkins

Ansible

Q170) What Are the Vagrant And Its Uses?

Vagrant used to virtual box as the hypervisor for virtual environments and in current scenario it is also supporting the KVM. Kernel-based Virtual Machine.

Vagrant is a tool that can created and managed environments for the testing and developing software. Devops Training Free Demo

Q171) How to Devops Is Helpful To Developers ?

To fix the bug and implements new features of the quickly. It provides to the clarity of communications among team members.

Q172) Name of The Popular Scripting Language Of the Devops ?

Python

Q173) List of The Agile Methodology Of the Devops?

DevOps is a process

Agile is the same as DevOps.

Separate group are framed.

It is problem solving.

Developers managing production

DevOps is the development-driven release management

Q174) Which Are The Areas of Devops Are Implemented?

Production Development

Creation of the productions feedback and its development

IT Operations development

Q175) The Scope For SSH ?

SSH is a Secure Shell which provides users with a secure, encrypted mechanism to log into systems and transfer files.

To log out the remote machine and worked on command line.

To secure encrypted of the communications between two hosts over an insecure network.

Q176) What Are The Advantages Of Devops With Respect To the Technical And Business Perspective?

Technical benefits

Software delivery is continuous.

Reduces Complexity in problems.

Faster approach to resolve problems

Manpower is reduced.

Business benefits

High rate of delivering its features

Stable operating environments

More time gained to Add values.

Enabling faster feature time to market

Q177) What Are The Core Operations Of the Devops In Terms Of the Development And Infrastructure ?

The core operations of DevOps

Application development

Code developing

Code coverage

Unit testing

Packaging

Deployment With infrastructure

Provisioning

Configuration

Orchestration

Deployment

Q178) What Are The Anti-patterns Of Devops?

A pattern is common usage usually followed. If a pattern of thecommonly adopted by others does not work for your organization and you continue to blindly follow it, you are essentially adopting an anti-pattern. There are myths about DevOps.

Some of them include

DevOps is a process

Agile equals DevOps

We need a separate DevOps group

Devops will solve all our problems

DevOps means Developers Managing Production

DevOps is Development-driven release management

DevOps is not development driven.

DevOps is not IT Operations driven.

We can’t do DevOps – We’re Unique

We can’t do DevOps – We’re got the wrong people

Q179) What are The Most Important Thing Devops Helps Us Achieve?

The most important thing that the DevOps helps us achieve is to get the changes into production as quickly as possible while that minimizing risks in software quality assurance and compliance. This is the primary objective of DevOps.

For example clear communication and better working relationships between teams i.e. both of the Ops team and Dev team collaborate together to deliver good quality software which in turn leads to higher customer satisfaction.

Q180) How Can Make a Sure New Service Is Ready For The Products Launched?

Backup System

Recovery plans

Load Balancing

Monitoring

Centralized logging

Q181) How to All These Tools Work for Together?

Given below is a generic logical of the flow where everything gets are automated for seamless delivery. However, its flow may vary from organization to the organization as per the requirement.

Developers develop the code and this source code is managed by Version Control System of the tools like Git etc.

Developers send to this code of the Git repository and any changes made in the code is committed to this Repository.

Jenkins pulls this code from the repository using the Git plugin and build it using tools like Ant or Maven.

Configuration managements tools like puppet deploys & provisions testing environment and then Jenkins releases this code on the test to environment on which testing is done using tools like selenium.

Once the code are tested, Jenkins send it for the deployment on production to the server (even production server are provisioned & maintained by tools like puppet).

After deployment Its continuously monitored by tools like Nagios.

Docker containers provides testing environment to the test the build features.

Q182) Which Are The Top Devops Tools?

The most popular DevOps tools are mentioned below

Git Version Control System tool

Jenkins Continuous Integration tool

Selenium Continuous Testing tool

Puppet, Chef, Ansible are Configuration Management and Deployment tools

Nagios Continuous Monitoring tool

Docker Containerization tool

Q183) How to Devops Different From the Agile / Sdlc?

Agile are the set of the values and principles about how to produce i.e. develop software.

Example if you have some ideas and you want to the turn those ideas into the working software, you can use the Agile values are principles as a way to do that. But, that software might only be working on developer's laptop or in a test environment. You want a way to quickly, easily and repeatably move that software into the production infrastructure, in a safe and simple way. To do that you needs are DevOps tools and techniques.

You can summarize by saying Agile of the software development methodology focuses on the development for software but DevOps on the other hand is responsible for the development as well as deployment of the software to the safest and most reliable way to the possible. Here's a blog that will give you more information of the evolutions of the DevOps.

Q184) What Is The Need For Devops?

According to me, this should start by explaining the general market trend. Instead of the releasing big sets of the features, companies are trying to see if small features can be transported to their customers through a series of the release trains. This have many advantages like quick feedback from the customers, better quality of the software etc. which in turn leads to the high customer satisfaction.

To achieve this, companies are required to

Increase deployment frequency

Lower failure rate of new releases

Shortened lead time between fixes

Faster mean time to recovery of the event of new release crashing

Q185) What is meant by Continuous Integration?

It's the development practice that requires developers to integrate code into a shared repository several times a day. Each check-in then verified by an automated build, allowing teams to the detect problems early.

Q186) Mention some of the useful plugins in Jenkins.

Below, I have mentioned some important are Plugins:

Maven 2 project

Amazon EC2

HTML publisher

Copy artifact

Join

Green Balls

Q187) What is Version control?

Its the system that records changes are the file or set of the files over time so that you can recall specific versions later.

Q188) What are the uses of Version control ?

Revert files back to a previous state. Revert to the entire project back to a previous state.

Compare changes over time.

See who last modified the something that might to be causing a problem.

Who introduced an issue and when.

Q189) What are the containers?

Containers are the of lightweight virtualization, heavier than 'chroot' but lighter than 'hypervisors'. They provide isolation among processes

Q190) What is meant by Continuous Integration?

It is a development practice that requires are developers to integrate code into the shared repository several times a day.

Q191) What's a PTR in DNS?

Pointer (PTR) record to used for the revers DNS (Domain Name System) lookup.

Q192) What testing is necessary to insure a new service is ready for production?

Continuous testing

Q193) What is Continuous Testing?

It is the process of executing on tests as part of the software delivery pipelines to obtain can immediate for feedback is the business of the risks associated with in the latest build.

Q194) What is Automation Testing?

Automation testing or Test Automation is a process of the automating that manual process to test the application/system under test.

Q195) What are the key elements of continuous testing?

Risk assessments, policy analysis, requirements traceabilities, advanced analysis, test optimisation, and service virtualisations

Q196) What are the Testing types supported by Selenium?

Regression testing and functional testing

Also Read>> Top Selenium Interview Questions & Answers

Q197) What is Puppet?

It is a Configuration Management tool which is used to the automate administration of the tasks.

Q198) How does HTTP work?

The HTTP protocol are works in a client and server model like most other protocols. A web browser using which a request is initiated is called as a client and a web servers software which are the responds to that request is called a server. World Wide Web Consortium of the Internet Engineering Task Force are two importants spokes are the standardization of the HTTP protocol.

Q199) Describe two-factor authentication?

Two-factors authentication are the security process in which the user to provides two means of the identification from separate categories of credentials.

Q200) What is git add?

adds the file changes to the staging area

Q201) What is git commit?

Commits the changes to the HEAD (staging area)

Q202) What is git push?

Sends the changes to the remote repository

Q203) What is git checkout?

Switch branch or restore working files

Q204) What is git branch?

Creates a branch

Q205) What is git fetch?

Fetch the latest history from the remote server and updates the local repo

Q206) What is git merge?

Joins two or more branches together

Q207) What is git pull?

Fetch from and integrate with another repository or a local branch (git fetch + git merge)

Q208) What is git rebase?

Process of moving or combining a sequence of commits to a new base commit

Q209) What is git revert?

To revert a commit that has already been published and made public

Q210 What is git clone?

Ans: clones the git repository and creates a working copy in the local machine

Q211) What is the difference between the Annie Playbook book and the characters?

Roles

The characters are a restructured entity of a play. Plays are on playbooks.

A set of functions to accomplish the specific role. Maps between hosts and roles.

Example: Common, Winners. Example: site.yml, fooservers.yml, webservers.yml.

Q212) How do I see all the ansible_ variables list?

By naturally collecting “facts” about the machines, these facts can be accessed in Playbooks and in templates. To see a list of all the facts about a computer, you can run a “setup” block as an ad hoc activity:

Ansible -m system hostname

It will print a dictionary of all the facts available for that particular host.

Q213) What is Doctor?

Docax is a container technology that connects your application and all its functions into the form of containers to ensure that you are running uninterrupted in any situation of your use.

Q214) What is the Tagore film?

Tucker is the source of the dagger container. Or in other words, dagger pictures are used to create containers.

Q215) What is the tooger container?

Dogger Container is a phenomenon of the film.

Q216) Do we consider Dev Devils as a smart way?

Of course, we !! The only difference between dynamic algorithms and DevObs is that the dynamic process is implemented for the development section and activates both DevOps development and functionality.

Q217) What are the benefits of using Git?

Data personality and copy

Get high

only one. A directory directory in the repository

High disk usage and network performance

Joint friendship

Git can use any kind of projects.

Q218) What is kernel?

A kernel, the software that can easily change the hardware interfaces of your computer.

Q219) What is the difference between grep -i and grep -v?

I accept this value

L) ls | grep -i docker

Dockerfile

docker.tar.gz

ls | grep -v docker

Desktop

Dockerfile

Documents

Downloads

You can not find anything with name docker.tar.gz

Q220) You can define a specific location for the file

This feature is generally used to give the server a replacement location. Let me tell you on the computer below and I want to create 1GB swap space,

dd if = / dev / zero = = / swapfile1 bs = 1G count = 1

Q221) What is the concept of sudo in Linux?

Pseudo is an application for Unix-and Linux-based systems that provide the ability to allow specific users to use specific system commands in the system's root level.

Q222) What is Jenkins pipe?

Jenkins pipeline (or simply "tube") is an additional package that supports and activates continuous delivery tube in Jenkins.

Q223) How to stop and restart the toxin container?

Stop container: stop container container ID

Reboot the Tucker Container now: Docer Re-container ID

Q224) Which sites are running by Tagore?

Docax is running on Linux and Cloud platforms only:

Ubuntu 12.04 LTS +

Fedora 20+

RHEL 6.5+

CentOS 6+

Gentoo

ArchLinux

openSUSE 12.3+

CRUX 3.0+

Cloud:

Amazon EC2

Google Compute Engine

Microsoft Asur

Rackspace

Since support is not supported, do not work on Windows or Mac for token production, yes, even on windows you can use it for testing purposes

Q225) What are the tools used for taxi networking?

We usually use karfs and taxi bear to do taxi networking.

Q226) What does Tucker write?

You would like to have a number of taxiers containers, and at that time you need to create a file that creates a docer and type the command to make a taxi-up. It runs all containers mentioned in the docer compose file.

Q227) What is a scrum?

Using scrime based on your complex software and product development task as small particles, it uses reboots and additional procedures. Each replay is two weeks. Scrum has three characters: product owner, scrum master and team

Q228) Purpose for SSH?

SSH is a secure shell that allows users to login to a secure, encrypted mechanism into computers and transmitting files.Exit the remote machine and work on the command line.

Protect encrypted communications between the two hosts on an unsafe network.

Q229) Are DevOps implemented?

Product development

Creating product feedback and its development

IT Activities Development.

Q230) Do you want to list the active modes of DevOps?

DevOps is a process

Like the active DevOps.

A separate group is configured.

This will solve the problem.

Manufacturers manufacturing production

DevOps is a development-driven output management

Q231) Do you list the main difference between active and DevOffice?

Agile:

There is something about dynamic software development

Devops:

DevOps is about software deployment and management.

DevOps does not replace the active or lean. By removing waste, by removing gloves and improving regulations, it allows the production of rapid and continuous products.

Q232) For the popular scripting language of DevOps?

Python

Q233) How does DevOps help developers?

To correct the defect and immediately make innovative attributes.

This is the accuracy of the coordination between the members of the group.

Q234) What is Vegand and its Uses?

Virtual virtual box has been used as a hyperversion for virtual environments and in the current scenario it supports KVM. Kernel-based virtual machine

Vegant is a tool for creating and managing the environment for making software and experiments. Tutorials Tutorial Free Demo

Q235) What is the main difference between Linux and Unix operating systems?

Unix:

It belongs to the multitasking, multiuser operating system family.

These are often used on web servers and workstations.

It was originally derived from AT & T Unix, which was started by the Bell Labs Research Center in the 1970s by Ken Thompson, Dennis Ritchie, and many others.

Operating systems are both open source, but the comparison is relatively similar to Unix Linux.

Linux:

Linux may be familiar to each programming language.

These personal computers are used.

The Unix operating system is based on the kernel.

Q236) How can we ensure how to prepare a new service for the products launched?

Backup system

Recovery plans

Load balance

Tracking

Centralized record

Q237) What is the benefit of NoSQL?

Independent and schema-less data model

Low latency and high performance

Very scalable

Q238) What is the adoption of Devokos in the profession

1. Use of active and other developmental processes and methods.

An increased ratio of production output is required from use and business.

3. Virtual and Cloud Infrastructure Transfers from Internal and Outdoor Providers;

4. Increased use of data center, automation and configuration management tools;

5. Focusing on testing automation and serial coordination systems;

6. Best Practices in Important Problems

Q239) What are the benefits of NoSQL database on RDBMS?

Benefits:

ETL is very low

Support for structured text is provided

Changes in periods are handled

Key Objectives Function.

The ability to measure horizontally

Many data structures are provided.

Vendors may be selected.

Q240) The first 10 capabilities of a person in the position of DevOp should be.

The best in system administration

Virtualization experience

Good technical skills

Great script

Good development skills

Chef in the automation tool experience

People management

Customer service

Real-time cloud movements

Who’s worried about who

Q241) What is PTR in DNS?

The PNS (PTR) registration is used to turn the search DNS (Domain Name System).

Q242) What do you know about DevOps?

Your answer should be simple and straightforward. Start by explaining the growing importance of DevOps in information technology. Considering that the efforts of the developments and activities to accelerate the delivery of software products should be integrated, the minimum failure rate. DevOps is a value-practical procedure in which the design and performance engineers are able to capture the product level or service life cycle across the design, from design and to the design level

Q243) Why was Dev’s so important in the past few years?

Before discussing the growing reputation of DevOps, discuss the current industry scenario. The big players like Netflix and Facebook begin with some examples of how this business can help to develop and use unwanted applications. Facebook’s continuous use and coding license models, and how to measure it, while using Facebook to ensure the quality of the experience. Hundreds of lines are implemented without affecting ranking, stability and security. Dipops Training Course

Your next application must be Netflix. This streaming and on-the-video video company follows similar procedures with complete automated processes and systems. Specify user base of these two companies: Facebook has 2 billion users, Netflix provides online content for more than 100 million users worldwide. Reduced lead time between the best examples of bugs, bug fixes, runtime and continuous supplies and the overall reduction of human costs.

Q244) What are some of the most popular DevOps tools?

The most popular DevOps tools include:

Selenium

Puppet

Chef

Git information

Jenkins

Ansible

Tucker Tipps Online Training

Q245) What is Version Control, and why should VCS use?

Define the control bar and talk about any changes to one or more files and store them in a centralized repository. VCS Tools remembers previous versions and helps to:

Make sure you do not go through changes over time.

Turn on specific files or specific projects to the older version.

Explore the problems or errors of a particular change.

Using VCS, developers provide flexibility to work simultaneously on a particular file, and all changes are logically connected.

Q246) Is There a Difference Between Active and DevOps? If yes, please explain

As a DevOps Engineer, interview questions like this are very much expected. Start by explaining the clear overlap between DevOps and Agile. Although the function of DevOps is always synonymous with dynamic algorithms, there is a clear difference between the two. Agile theories are related to the soft product or development of the software. On the other hand, DevOps is handled with development, ensuring quick turnaround times, minimal errors and reliability by installing the software continuously.

Q247) Why are structural management processes and tools important?

Talk about many software developments, releases, edits and versions for each software or testware. Describe the need for data storage and maintenance, development of developments and tracking errors easily. Do not forget to mention key CM tools that can be used to achieve these goals. Talk about how the tools, such as buffet, aseat, and chef are useful in automating software deployment and configuration on multiple servers.

Q248) How is the chef used as a CM tool?

Chef is considered one of the preferred professional CM Tools. Facebook has changed its infrastructure and the Shef platform keeps track of IT, for example. Explain how the chef helps to avoid delays by automating processes. The scripts are written in ruby. It can be integrated into cloud-based platforms and configures new settings. It provides many libraries for infrastructure development, which will then be installed in a software. Thanks to its centralized management system, a chef server is sufficient to use various policies as the center of ordering.

Q249) How do you explain the concept of “Infrastructure Index” (IAC)?

This is a good idea to talk about IAC as a concept, sometimes referred to as a programming program, where the infrastructure is similar to any other code. The traditional approach to managing infrastructure is how to take a back seat and how to handle manual structures, unusual tools and custom scripts

Q250) List the essential DevOps tools.

Git information

Jenkins

Selenium

Puppet

Chef

Ansible

Nagios

Laborer

Monit

El-Elistorsch, Lestastash, Gibbon

Collectd / Collect

Git Information (Gitwidia)

Q251) What are the main characters of DevOps engineers based on growth and infrastructure?

DevOps Engineer’s major work roles

Application Development

Developing code

Code coverage

Unit testing

Packaging

Preparing with infrastructure

Continuous integration

Continuous test

Continuous sorting

Provisioning

Configuration

Orchestration

Deployment

Q252) What are the advantages of DevOps regarding technical and business perspective?

Technical Advantages:

Software delivery continues.

Problems reduce austerity.

Fast approach to solving problems

Humans are falling.

Business Benefits:

The higher the rate for its features

Fixed operating systems

It took too long to add values.

Run fast time for the market

Learn more about DevOps benefits from this information blog.

Q253) Purpose for SSH?

SSH is a secure shell that allows users to login to a secure, encrypted mechanism into computers and transmitting files.

Exit the remote machine and work on the command line.

Protect encrypted communications between the two hosts on an unsafe network.

Q254) Which part of DevOps is implemented?

Product development

Creating product feedback and its development

IT Activities Development

Q255) List the DevOps's active algorithm.

DevOps is a process

Like the active DevOps.

A separate group is configured.

This will solve the problem.

Manufacturers manufacturing production

DevOps is a development-driven output management

Q256) List the main difference between active and devOps.

Agile:

There is something about dynamic software development

Devops:

DevOps is about software deployment and management.

DevOps does not replace the active or lean. By removing waste, by removing gloves and improving regulations, it allows the production of rapid and continuous products.

Q257) For the popular scripting language of DevOps.

Python

Q258) How does DevOps help developers?

Correct the error and activate new features quickly.

It provides clarity of clarity between the members of the group.

Q259) What is the speed and its benefits?

Virtual virtual box has been used as a hyperversion for virtual environments and in the current scenario it supports KVM. Kernel-based virtual machine

Vegant is a tool for creating and managing the environment for making software and experiments.

Q260) What is the use of Anuj?

It is mainly used for information technology infrastructure to manage or use applications for remote applications. We want to sort an app on the nodes of 100 by executing one command, then the animation is actually in the picture, but you need to know or run some knowledge on the animated script.

Q1.What is Infrastructure as Code?

Answer: Where the Configuration of any servers or toolchain or application stack required for an organization can be made into more descriptive

level of code and that can be used for provisioning and manage infrastructure elements like Virtual Machine, Software, Network Elements,

but it differs from scripts using any language, where they are series of static steps coded, where Version control can be used in order

to track environment changes.Example Tools are Ansible, Terraform.

Q2.What are the areas the Version control can introduce to get efficient DevOps practice?

Answer: Obviously the main area of Version Control is Source code management, Where every developer code should be pushed to the common repository for maintaining build and release in CI/CD pipelines.Another area can be Version control For Administrators when they use Infrastructure as A Code (IAC) tools and practices for maintaining The Environment configuration.Another Area of Version Control system Can be Artifactory Management Using Repositories like Nexus & DockerHub.</p>

Q3.Why the Opensource tools boost DevOps?

Answer: Opensource tools predominantly used by any organization which is adapting (or) adopted DevOps pipelines because devops came with the focus of automation in various aspects of organization build and release and change management and also infrastructure management areas. So developing or using a single tool is impossible and also everything is basically trial and error phase of development and also agile cuts down the luxury of developing a single tool , so opensource tools were available on the market pretty much saves every purpose and also gives organization an option to evaluate the tool based on their need .

Q4.What is the difference between Ansible and chef(or) puppet?

Answer: Ansible is Agentless configuration management tool, where puppet or chef needs agent needs to be run on the agent node and chef or puppet is based on pull model, where your cookbook or manifest for chef and puppet respectively from the master will be pulled by the agent and ansible uses ssh to communicate and it gives data-driven instructions to the nodes need to be managed , more like RPC execution, ansible uses YAML scripting, whereas puppet (or) chef is built by ruby uses their own DSL .

Q5.What is folder structure of roles in ansible?

Answer: roles common tasks handlers files templates vars defaults meta webservers tasks defaults meta/ Where common is role name, under tasks – there will be tasks (or) plays present, handlers – to hold the handlers for any tasks, files – static files for copying (or) moving to remote systems, templates- provides to hold jinja based templating , vars – to hold common vars used by playbooks.

Q6. What is Jinja2 templating in Ansible playbooks and their use?

Answer: Jinja2 templating is the Python standard for templating , think of it like a sed editor for Ansible , where it can be used is when there is a need for dynamic alteration of any config file to any application like consider mapping a MySQL application to the IP address of the machine, where it is running, it cannot be static , it needs altering it dynamically at runtime . Format {{ foo.bar }} The vars within the {{ }} braces are replaced by ansible while running using template module.

Q7. What is the need for organizing playbooks as the role, is it necessary?

Answer: Organizing playbooks as roles , gives more readability and reusability to any plays , while consider a task where MySQL installation should be done after the removal of Oracle DB , and another requirement is needed to install MySQL after java installation, in both cases we need to install MySQL , but without roles need to write playbooks separately for both use cases , but using roles once the MySQL installation role is created can be utilised any number of times by invoking using logic in site.yaml . No, it is not necessary to create roles for every scenario, but creating roles is a best practice in Ansible.

Q8.What is main disadvantage of Docker containers?

Answer: As the lifetime of any containers is while running after a container is destroyed you cannot retrieve any data inside a container, the data inside a container is lost forever, but persistent storage for data inside containers can be done using volumes mount to an external source like host machine and any NFS drivers.</p>

Q9. What are docker engine and docker compose?

Answer: Docker engine contacts the docker daemon inside the machine and creates the runtime environment and process for any container, docker composes links several containers to form as a stack used in creating application stacks like a LAMP, WAMP, XAMP.</p>

Q10. What are the Different modes does a container can be run?

Answer: Docker container can be run in two modes Attached: Where it will be run in the foreground of the system you are running, provides a terminal inside to container when -t option is used with it, where every log will be redirected to stdout screen. Detached: This mode is usually run in production, where the container is detached as a background process and every output inside the container will be redirected log files inside /var/lib/docker/logs/<container-id>/<container-id.json> and which can be viewed by docker logs command.

Q11. What the output of docker inspect command will be?

Answer: Docker inspects <container-id> will give output in JSON format, which contains details like the IP address of the container inside the docker virtual bridge and volume mount information and every other information related to host (or) container specific like the underlying file driver used, log driver used. docker inspect [OPTIONS] NAME|ID [NAME|ID…] Options
 Name, shorthand Default Description
 — format, -f Format the output using the given Go template
 –size , -s Display total file sizes if the type is container
 –type Return JSON for specified type

Q12.What is the command can be used to check the resource utilization by docker containers?

Answer: Docker stats command can be used to check the resource utilization of any docker container, it gives the output analogous to Top command in Linux, it forms the base for container resource monitoring tools like advisor, which gets output from docker stats command. docker stats [OPTIONS] [CONTAINER…] Options
 Name, shorthand Default Description
 — all, -a Show all containers (default shows just running)
 –format Pretty-print images using a Go template
 –no-stream Disable streaming stats and only pull the first result
 –no-trunc Do not truncate output

Q13.What is the major difference between Continuos deployment and continuos delivery?

Answer: Where continuos deployment is fully automated and deploying to production needs no manual intervention in continuos deployment, whereas in continuos delivery the deployment to production has some

manual intervention for change management in Organization for better management, and it needs to approved by manager or higher authorities to be deployed in production. According to your application risk factor for organization, the continuos deployment (or) delivery approach will be chosen.

Q14.How to execute some task (or) play on localhost only while executing playbooks on different hosts on an ansible?

Answer: In ansible, there is a module called delegate_to, in this module section provide the particular host (or) hosts where your tasks (or) task need to be run.

tasks:

– name: ” Elasticsearch Hitting”

uri: url='{{ url2 }}_search?

Q=status:new’ headers='{“Content-type”:”application/json”}’ method=GET return_content=yes

register: output

delegate_to: 127.0.0.1

Q15. What is the difference between set_fact and vars in ansible?

Answer: Where a set_fact sets the value for a factor at one time and remains static, even though the value is

Quite dynamic and vars keep on changing as per the value keeps on changing for the variable. tasks: – set_fact: fact_time: “Fact: {{lookup(‘pipe’, ‘date \”+%H:%M:%S\”‘)}}” – debug: var=fact_time – command: sleep 2 – debug: var=fact_time tasks: – name: lookups in variables vs. lookups in facts hosts: localhost vars: var_time: “Var: {{lookup(‘pipe’, ‘date \”+%H:%M:%S\”‘)}}” Even though the lookup for date has been used in both the cases , where in the vars is used it alters based on the time to time every time executed within the playbook lifetime. But Fact always remains same once lookup is done

Q16. What is the lookup in ansible and what are lookup plugins supported by ansible?

Answer: Lookup plugins allow access of data in Ansible from outside sources. These plugins are evaluated on the Ansible control machine, and can include reading the filesystem but also contacting external datastores and services. Format is {lookup{‘<plugin>’,’<source(or)connection_string>’}} Some of the lookup plugins supported by ansible are File pipe redis jinja templates etcd kv store …

Q17. How can you delete the docker images stored on your local machine and how can you do it for all the images at once?

Answer: The command docker rmi <image-id> can be used to delete the docker image from local machine , whereas some images may need to be forced because the image may be used by some other container (or) another image , to delete images you can use the combination of commands by docker rmi $(docker images -

Q) , where docker images will give the docker image names , to get only the ID of docker images only , we are using -

Q switch with docker images command.

Q18. What are the folders in the Jenkins installation and their uses?

Answer: JENKINS_HOME – which will be /$JENKINS_USER/.jenkins it is the root folder of any Jenkins installation and it contains subfolders each for different purposes. jobs/ – Folder contains all the information about all the jobs configured in the Jenkins instance. Inside jobs/, you will have the folder created for each job and inside those folders, you will have build folders according to each build numbers each build will have its log files, which we see in Jenkins web console. Plugins/ – where all your plugins will be listed. Workspace/ – this will be present to hold all the workspace files like your source code pulled from SCM.

Q19. What are the ways to configure Jenkins system?

Answer: Jenkins can be configured in two ways Web: Where there is an option called configure system , in there section you can make all configuration changes . Manual on filesystem: Where every change can also be done directly on the Jenkins config.xml file under the Jenkins installation directory , after you make changes on the filesystem, you need to restart your Jenkins, either can do it directly from terminal (or) you can use Reload configuration from disk under manage Jenkins menu or you can hit /restart endpoint directly.

Q20. What is the role Of HTTP REST API in DevOps?

Answer: As Devops is purely focuses on Automating your infrastructure and provides changes over the pipeline for different stages like an each CI/CD pipeline will have stages like build,test,sanity test,UAT,Deployment to Prod environment as with each stage there are different tools is used and different technology stack is presented and there needs to be a way to integrate with different tool for completing a series toolchain, there comes a need for HTTP API , where every tool communicates with different tools using API , and even user can also use SDK to interact with different tools like BOTO for Python to contact AWS API's for automation based on events , nowadays its not batch processing anymore , it is mostly event driven pipelines

Q21. What are Microservices, and how they power efficient DevOps practices?

Answer: Where In traditional architecture , every application is monolith application means that anything is developed by a group of developers , where it has been deployed as a single application in multiple machines and exposed to outer world using loadbalancers , where the microservices means breaking down your application into small pieces , where each piece serves the different functionality needed to complete a single transaction and by breaking down , developers can also be formed to groups and each piece of application may follow different guidelines for efficient development phase , because of agile development should be phased up a bit and every service uses REST API (or) Message

Queues to communicate between other service. So build and release of a non-robust version may not affect whole architecture , instead some functionality is lost , that provides the assurance for efficient and faster CI/CD pipelines and DevOps Practices

Q22. What are the ways that a pipeline can be created in Jenkins?

Answer: There are two ways of the pipeline can be created in Jenkins Scripted Pipelines: More like a programming approach Declarative pipelines: DSL approach specifically for creating Jenkins pipelines. The pipeline should be created in Jenkins file and the location can either be in SCM or local system. Declarative and Scripted Pipelines are constructed fundamentally differently. Declarative Pipeline is a more recent feature of Jenkins Pipeline which: Provides richer syntactical features over Scripted Pipeline syntax, and is designed to make writing and reading Pipeline code easier.

Q23. What are the Labels in Jenkins & where it can be utilised?

Answer: As with CI/CD solution needs to be centralized , where every application in the organization can be built by a single CI/CD server , so in organization there may be different kinds of application like java , c#,.NET and etc , as with microservices approach your programming stack is loosely coupled for the project , so you can have Labels in each node and select the option Only built jobs while label matching this node , so when a build is scheduled with the label of the node present in it , it waits for next executor in that node to be available , eventhough there are other executors in nodes.

Q24. What is the use of Blueocean in Jenkins?

Answer: Blue Ocean rethinks the user experience of Jenkins. Designed from the ground up for Jenkins Pipeline, but still compatible with freestyle jobs, Blue Ocean reduces clutter and increases clarity for every member of the team. It provides sophisticated UI to identify each stage of the pipeline and better pinpointing for issues and very rich Pipeline editor for beginners.

Q25. What is the callback plugins in ansible, give some examples of some callback plugins?

Answer: Callback plugins enable adding new behaviors to Ansible when responding to events. By default, callback plugins control most of the output you see when running the command line programs, but can also be used to add additional output, integrate with other tools and marshall the events to a storage backend. So whenever an play is executed and after it produces some events , that events are printed onto Stdout screen ,so callback plugin can be put into any storage backend for log processing. Example callback plugins are ansible-logstash, where every playbook execution is fetched by logstash in the JSON format and can be integrated any other backend source like elasticsearch.

Q26. What are the scripting languages can be used in DevOps?

Answer: As with scripting languages , the basic shell scripting is used for build steps in Jenkins pipelines and python scripts can be used with any other tools like Ansible , terraform as a wrapper script for some other complex decision solving tasks in any automation as python is more superior in complex logic derivation than shell scripts and ruby scripts can also be used as build steps in Jenkins.

Q27. What is Continuos Monitoring and why monitoring is very critical in DevOps?

Answer: Devops brings out every organization capablity of build and release cycle to be much shorter with concept of CI/CD , where every change is reflected into production environments fastly , so it needs to be tightly monitored to get customer feedbacks. So the concept of continuos monitoring has been used to evaluate each application performance in real time (atleast Near Real Time) , where each application is developed with application performance monitor agents compatible and the granular level of metrics are taken out like JVM stats and even fuctional wise metrics inside the application can also be poured out in real time to Agents , which in turn gives to any backend storage and that can be used by monitoring teams in dashboards and alerts to get continuosly monitor the application

Q28. Give some examples of continuos monitoring tools?

Answer: Where many continuos monitoring tools are available in the market, where used for a different kind of application and deployment model Docker containers can be monitored by cadvisor agent , which can be used by Elasticsearch to store metrics (or) you can use TICK stack (Telegraf, influxdb,Chronograf,Kapacitor) for every systems monitoring in NRT(Near Real Time) and You can use Logstash (or) Beats to collect Logs from system , which in turn can use Elasticsearch as Storage Backend can use Kibana (or) Grafana as visualizer. The system monitoring can be done by Nagios and Icinga.

Q29. What is docker swarm?

Answer: Group of Virtual machines with Docker Engine can be clustered and maintained as a single system and the resources also being shared by the containers and docker swarm master schedules the docker container in any of the machines under the cluster according to resource availability Docker swarm init can be used to initiate docker swarm cluster and docker swarm join with the master IP from client joins the node into the swarm cluster.

Q30. What are the ways to create Custom Docker images?

Answer: Docker images can be created by two ways broadly Dockerfile: Most used method , where base image can be specified and the files can be copied into the image and installation and configuration can be done using declarative file which can be given to Docker build command to produce new docker image. Docker commit: Where the Docker image is pinned up as a Docker container and every command execute inside a container forms a Read-only layer and after every changes is Done can use docker commit <container-iD> to save as a image, although this method is not suitable for CI/CD pipelines , as it re

Quires manual intervention.

Q31. Give some important directives in Dockerfile and an example Dockerfile?

Answer: FROM – Gives the base image to use. RUN – this directive used to run a command directly into any image. CMD- To run the command, but the format of command specification is more arguments based than a single command like RUN. ADD (or) COPY – To copy files from your local machine to Docker images you create. ENTRYPOINT- Entrypoint command keeps the command without execution, so when the container is spawned from an image, the command in entry point runs first. Example Dockerfile

FROM python:2

MAINTAINER janakiraman

RUN mkdir /code

ADD test.py /code

ENTRYPOINT [“python”,”/code/test.py”]

Q32. Give some important Jenkins Plugins

Answer: SSH slaves plugin
Pipeline Plugin
Github Plugin
Email notifications plugin
Docker publish plugin
Maven plugin
Greenball plugin

Q33.What is the use of vaults in ansible?

Answer: Vault files are encrypted files, which contains any variables used by ansible playbooks, where the vault encrypted files can be decrypted only by the vault-password, so while running a playbook, if any vault file is used for a variable inside playbooks, so need to used –-ask-vault-pass command argument while running playbook.

Q34. How does docker make deployments easy ?

Answer: Docker is a containerization technology, which is a advanced technology over virtualization, where in virtualization, an application needs to be installed in machine , then the OS should be spin up and spinning up Virtual machine takes lot time , and it divides space from Physical hardware and hypervisor layer wastes vast amount of space for running virtual machines and after it is provisioned, Every application needs to be installed and installation re

Quires all dependencies and sometimes dependencies may miss out even if you double check and migration from machine to machine of applications is painful , but docker shares underlying OS resources , where docker engine is lightweight and every application can be packaged with dependency once tested works everywhere same, migration of application or spinning up of new application made easy because just needs to install only docker in another machine and docker image pull and run does all the magic of spinning up in seconds.

Q35. How .NET applications can be built using Jenkins?

Answer: .NET applications needs Windows nodes to built , where Jenkins can use Jenkins windows slave plugin can be used to connect windows node as a Jenkins slave , where it uses DCOM connector for Jenkins master to slave connection (or) you can use Jenkins JNLP connector and the Build tools and SCM

tools used for the pipeline of .NET application needs to be installed in the Windows slave and MSBuild build tool can be used to build .NET application and can be Deployed into Windows host by using Powershell wrapper inside Ansible playbooks.

Q36. How can you make a High available Jenkins master-master solution without using any Jenkins plugin?

Answer: Where Jenkins stores all the build information in the JENKINS_HOME directory , which can be mapped to any NFS (or) SAN storage drivers , common file systems and when the node is down , can implement a monitoring solution using Nagios to check alive , if down can trigger an ansible playbook (or) python script to create a new Jenkins master in different node and reload at runtime, if there is already a passive Jenkins master in another instance kept silent with same JENKINS_HOME Network file store.

Q37. Give the structure of Jenkins file?

Answer: Jenkins filed starts with Pipeline directive , inside the pipeline directive will be agent directive , which specifies where the build should be run and next directive would be stages , which contains several list of stage directives and each stage directive contains different steps . There are several optional directives like options , which provides custom plugins used by the projects (or) any other triggering mechanisms used and environment directive to provide all env variables Sample Jenkins file pipeline{ agent any stages { stage('Dockerbuild') { steps { sh "sudo docker build. -t pyapp:v1" } } } }

Q38. What are the uses of integrating cloud with DevOps?

Answer: The centralized nature of cloud computing provides DevOps automation with a standard and centralized platform for testing, deployment, and production.Most cloud providers gives Even DevOps technologies like CI tools and deployment tools as a service like codebuild, codepipeline, codedeploy in AWS makes easy and even faster rate of DevOps pratice.

Q39. What is Orchestration of containers and what are the different tools used for orchestration?

Answer: When deploying into production, you cannot use a single machine for production as it is not robust for any deployment , so when an application is containerized, the stack of applications maybe run at single docker host in development environment to check application functionality, while when we arrive into production servers, that it is not the case, where you should deploy your applications into multiple nodes and stack should be connected between nodes , so to ensure network connectivity between different containers , you need to have shell scripts (or) ansible playbooks between different nodes ,and another disadvantage is using this tools , you cannot run an efficient stack, where an application is taking up more resources in one node , but another sits idle most time , so deployment strategy also needs to be planned out according to resources and load-balancing of this applications also be configured, so to clear out all this obstacles , there came a concept called orchestration , where your docker containers is orchestrated between different nodes in the cluster based on resources available according to scheduling strategy and everything should be given as DSL specific files not like scripts .There are Different Orchestration tools available in market which are Kubernetes,Swarm,Apache Mesos.

Q40. What is ansible tower?

Answer: Ansible is developed by Redhat , which provides IT automation and configuration management purposes. Ansible Tower is the extended management layer created to manage playbooks organization using roles and execution and can even chain different number of playbooks to form workflows. Ansible tower dashboard provides NOC-style UI to look into the status of all ansible playbooks and hosts status.

Q41. What are the programming language applications that can be built by Jenkins?

Answer: Jenkins is a CI/CD tool not depends on any Programming language for building application, if there is a build tool to built any language, that's enough to build, even though plugin for build tool not available, can use any scripting to replace your build stage like Shell, Powershell, Python scripts to make build of any language application.

Q42. Why is every tool in DevOps is mostly has some DSL (Domain Specific Language)?

Answer: DevOps is culture developed to address the needs of agile methodology , where the developement rate is faster ,so deployment should match its speed and that needs operations team to co-ordinate and work with dev team , where everything can automated using script-based , but it feels more like operations team than , it gives messy organization of any pipelines , more the use cases , more the scripts needs to be written , so there are several use cases, which will be ade

Quate to cover the needs of agile are taken and tools are created according to that and customiztion can happen over the tool using DSL to automate the DevOps practice and Infra management.

Q43. What are the clouds can be integrated with Jenkins and what are the use cases?

Answer: Jenkins can be integrated with different cloud providers for different use cases like dynamic Jenkins slaves, Deploy to cloud environments. Some of the clouds can be integrated are AWS
Azure
Google Cloud
OpenStack

Q44. What are Docker volumes and what type of volume should be used to achieve persistent storage?

Answer: Docker volumes are the filesystem mount points created by user for a container or a volume can be used by many containers , and there are different types of volume mount available empty dir, Post mount, AWS backed lbs volume, Azure volume, Google Cloud (or) even NFS, CIFS filesystems, so a volume should be mounted to any of the external drive to achieve persistent storage , because a lifetime of files inside container , is till the container is present and if container is deleted, the data would be lost.

Q45. What are the Artifacts repository can be integrated with Jenkins?

Answer: Any kind of Artifacts repository can be integrated with Jenkins, using either shell commands (or) dedicated plugins, some of them are Nexus, Jfrog.

Q46. What are the some of the testing tools that can be integrated with jenkins and mention their plugins?

Answer: Sonar plugin – can be used to integrate testing of Code

Quality in your source code. Performance plugin – this can be used to integrate JMeter performance testing. Junit – to publish unit test reports. Selenium plugin – can be used to integrate with selenium for automation testing.

Q47. What are the build triggers available in Jenkins?

Answer: Builds can be run manually (or) either can automatically triggered by different sources like Webhooks– The webhooks are API calls from SCM , whenever a code is committed into repository (or) can be done for specific events into specific branches. Gerrit code review trigger– Gerrit is an opensource code review tool, whenever a code change is approved after review build can be triggered. Trigger Build Remotely – You can have remote scripts in any machine (or) even AWS lambda functions (or) make a post re

Quest to trigger builds in Jenkins. Schedule Jobs- Jobs can also schedule like Cron jobs. Poll SCM for changes – Where your Jenkins looks for any changes in SCM for given interval, if there is a change, the build can be triggered. Upstream and Downstream Jobs– Where a build can be triggered by another job that is executed previously.

Q48. How to Version control Docker images?

Answer: Docker images can be version controlled using Tags , where you can assign tag to any image using docker tag <image-id> command. And if you are pushing any docker hub registry without tagging the default tag would be assigned which is latest , even if a image with the latest is present , it demotes that image without tag and reassign that to the latest push image.

Q49. What is the use of Timestamper plugin in Jenkins?

Answer: It adds Timestamp to every line to the console output of the build.

Q50.Why should you not execute a build on master?

Answer: You can run a build on master in Jenkins , but it is not advisable , because the master already has the responsibility of scheduling builds and getting build outputs into JENKINS_HOME directory ,so if we run a build on Jenkins master , then it additionally needs to build tools, and workspace for source code , so it puts performance overload in the system , if the Jenkins master crashes , it increases the downtime of your build and release cycle.

Q51. Why devops?

Answer: DevOps is the market trend now, which follows a systematic approach for getting the application live to market. DevOps is all about tools which helps in building the development platform as well as production platform. Product companies are now looking at a Code as a service concept in which the development skill is used to create a production architecture with atmost no downtime.

Q52. Why Ansible?

Answer: A Configuration Management tool which is agentless. It works with key based or password based ssh authentication. Since it is agentless, we have the complete control of the manipulating data. Ansible is also use for architecture provisioning as it has modules which can talk to major cloud platforms. I have mainly used for AWS provisioning and application/system config manipulations.

Q53. Why do you think a Version control system is necessary for DevOps team?

Answer: Application is all about code, if the UI is not behaving as expected, there could be a bug in the code. Inorder to track the code updates, versioning is a must.

By any chance if bug breaks the application, we should be able to revert it to the working codebase. Versioning helps to achieve this.

Also, by keeping a track of code commits by individuals, it is very easy to find the source of the bug in the code.

Q54. What role would you prefer to be in the DevOps team?

Answer: Basically the following are prominent in DevOps depending upon the skillset.

1. Architect
2. Version Control Personnel
3. Configuration control Team
4. Build and Integration management
5. Deployment Team.
6. Testing People
7.

QA

Q55. Architecture Monitoring Team

Answer: In my opinion, everyone should owe to be an architech. with this course, I will be fir the role from 2 to 5. Everyone should understand the working of each role. Devops is a collective effort rather individual effect.

Q56. Suppose you are put in to a project where you have to implement devops culture, what will be your approach?

Answer: Before thinking of DevOps, there should be a clear cut idea on what need to be implement and it should be done by the Senior architect.

If we take a simple example of shopping market :

Output of this business will be a website which displays online shopping items, and a payment platform for easy payment.

Even though it looks simple, the background work is not that easy, because a shopping cart must be :

– 99.99% live

– Easy and fast processing of shopping items

– Easy and fast payment system.

–

Quick reporting to shopkeeper

–

Quick Inventory Management

– Fast customer interaction

and many more

DevOps has to be implement in each process and phase. Next is the tools used for bringing the latest items in website with minimal time span. Git, Jenkins, Ansible/Chef, AWS can be much of familiar tools with helps in continuous delivery to market.

Q57. Whether continuous deployment is possible practically?

Answer: Ofcourse it is possible if we bring the Agility in every phase of development and deployment. The release, testing and deployment automation should be so accurately finetuned

Q58. What is Agility in devops basically?

Answer: Agile is an iterative form of process which finalizes the application by fulfilling the checklist. For any process, there should be set of checklist inorder to standardize the code as well as the build and deployment process. The list depends on the architecture of the application and business model.

Q59. Why scripting using Bash, Python or any other language is a must for a DevOps team?

Answer: Even though we have numerous tools in devops, but there will certain custom re

Quirements for a project. In such cases, we have to make use of scripting and then integrate it with the tools.

Q60. In AWS, how do you implement high availability of websites?

The main concept of high availability is that the website should be live all the time. So we should avoid single point of failure, inorder to achieve this LoadBalancer can be used. In AWS, we can implement HA with LB with AutoScaling methods.

Q61.How to debug inside a docker container ?

Answer: The feature “docker exec” allows users to debug a container

Q62.What do you mean by Docker Engine ?

It is open source container build and management tool

Q63.Why we need Docker?

Answer: Applications were started to use Agile methodology where they build and deployed iteratively .

Docker helps is deploying same binaries with dependencies across different environments with fraction of seconds

Q64.What do you mean by Docker daemon ?

Answer: Docker Daemon Receives and processes incoming API re

Quests from the CLI .

Q65.What do you mean by Docker client ?

Answer: Command line tool – which is a docker binary and it communicate to the Docker daemon through the Docker API.

Q66.what do you mean by Docker Hub Registry ?

Answer: It is a Public image registry maintanined by Docker itself and the Docker daemon talks to it through the registry API

Q67.How do you install docker on a debian Linux OS ?

Answer: sudo apt-get install docker.io

Q68.What access does docker group have ?

Answer: The docker user have root like access and we should restrict access as we would protect root

Q69.How to list the packages installed in Ubuntu container ?

Answer: dpkg -l lists the packages installed in ubuntu container

Q70.How can we check status of the latest running container?

Answer: With “docker ps -l” command list latest running processes

Q71.How to Stop a container?

Answer: “docker kill “command to kill a container “docker stop “command to stop a container

Q72.How to list the stopped containers?

Answer: docker ps -a (–an all)

Q73.What do you mean by docker image?

Answer: An image is a collection of files and its meta data , basically those files are the root filesystem of the container Image is made up of layers where each layer can be edited

Q74.What is the differences between containers and images

Answer: An image is an read-only filesystem where container is a running form of an image .

Image is non-editable and on containers we can edit as we wish & save that again to a new image

Q75.How to do changes in a docker image?

Answer: No we can’t do changes in an image. we can make changes in a Dockerfile or to the existing container to create a layered new image

Q76.Different ways to create new images ?

Answer: docker commit: to create an image from a container

docker build: to create an image using a Dockerfile

Q77.Where do you store and manage images?

Answer: Images can be stored in your local docker host or in a registry .

Q78.How do we download the images?

Answer: Using “docker pull” command we can download a docker image

Q79. What are Image tags?

Answer: Image tags are variants of Docker image . “latest” is the default tag of an image

Q80.What is a Dockerfile.?

Answer: A Dockerfile series of instructions to build a docker image Docker build command can be used to build

Q81.How to build a docker file?

Answer: docker build -t <image_name>

Q82.How to view hostory of a docker image?

Answer: The docker history command lists all the layers in an image with image creation date, size and command used

Q83.What are CMD and ENTRYPOINT?

Answer: These will allow using the default command to be executed when a container is starting

Q84.EXPOSE instruction is used for?

Answer: The EXPOSE command is used to publish ports of a docker container

Q85.What is Ansible?

Answer: A configuration management tool similar to a puppet , chef etc .

Q86.Why to choose Ansible?

Answer: Ansible is simple and light where it needs only shh and python as a dependency . It doesnt re Quired an agent to be installed

Q87.What are the ansible modules?

Answer: Ansible “modules” are pre-defined small set of codes to perform some actions eg: copy a file, start a service

Q88.What are Ansible Tasks ?

Answer: Tasks are nothing but ansible modules with the arguments

Q89.What are Handlers in ansible?

Answer: Handlers are triggered when there is need in change of state e.g.restart service when a property file have changed.

Q90.What are Roles in ansible?

Answer: Roles are re-usable tasks or handlers.

Q91.What is YAML?

Answer: YAML – yet another markup language is way of storing data in a structured text format like JSON

Q92.What are Playbooks ?

Answer: Playbooks are the recipes to ansible

Q93.What is MAVEN ?

Answer: Maven is a Java build tool, so you must have Java installed to proceed.

Q94.What do you mean by validate in maven ?

Answer: Validate is to check whether the info provided are correct and all necessary is available

Q95.What do you mean by compile in maven?

Answer: It is to compile the source code of the project

Q96.What do you mean by test in maven?

Answer: It is to test the source code to test using suitable testing framework

Q97.What do you mean by package in maven?

Answer: It is to do the binary packaging of the compiled code

Q98.What is docker-compose?

Answer: Compose is to define and run a multi-container application

Q99.What is Continuous integration?

Answer: CI is nothing but giving immediate feedback to the developer by testing , analyzing the code .

Q100. What is Continuous delivery?

Answer: Continuous delivery is a continuation of CI which aims in delivering the software until pre -prod automatically

Q101.What is Continuous deployment?

Answer: Continuous deployment is next step after CI and CD where the tested software will be provide to the end customers post some validation and change management activities

Q102.What is git?

Answer: git is a source code version management system .

Q103.What is git commit?

Answer: git commit records changes done to file in the local system.

Q104.what is git push?

Answer: git push is to update the changes to the remote repository in the internet .

Q105.What’s git fetch?

git fetch will pull only the data from the remote repo but doesnt merge with the repo in your local system.

Q106.What is git pull?

Answer: git pull will download the files from the remote repo and will merge with the files in your local system.

Q107.How to reset the Last git Commit ?

Answer: “git reset” command can be used to undo last commit .

Q108.What is the need for DevOps ?

Answer: Start the answer by explaining general market trend, how releasing small features benefits compared to releasing big features, advantages of releasing small features in high fre

Quency. Discuss about the topics such as Increase deployment fre

Quency
Lower failure rate of newer releases
Reduced time for bug fixes
Time to recovery

Q109. Write the key components of DevOps?

Answer: These are te key comonents of DevOps. Continuous Integration
Continuous Testing
Continuous Delivery
Continuous Monitoring

Q110. What are the various tools used in DevOps?

Answer: DevOps contains various stages. Each stage can be achieved with various tools. Below are the various tool that are popularly used tools in DevOps. Version Control : Git , SVN
CI/CD : Jenkins
Configuration Management Tools : Chef, Puppet, Ansible
Containerization Tool : Docker Also mention any other tools that you worked on that helped you to automate the existing environment

Q111. What is Version Control?

Answer: Version Control System (that are made to the files or documents over a period of time.

Q112. What are the types of Version Control Systems?

Answer: There are two types of Version Control Systems: Central Version Control System, Ex: Git, Bitbucket
Distributed/Decentralized Version Control System

Q113. What is jenkins?In jenkins, what is the programming language should be used?

Answer: It is a open Source automation tool. it is a pupose of Continuous Integration and Continuous Delivery. Jenkins is a written in java Programming language.

Q114. Give an explanation about DevOps.

Answer: DevOps is nothing but a practice that emphasizes the collaboration and communication of both software developers and implementation team. It focuses on delivering software product faster and lowering the failure rate of releases.

Q115. What are the key Principles or Aspects behind DevOps?

Answer: The key Principles or Aspects are Infrastructure as code
Continuous deployment
Automation
Monitoring
Security

Q116. Describe the core operations of DevOps with Infrastructure and with application.

Answer: The core operations of DevOps are Infrastructure Provisioning
Configuration
Orchestration
Deployment Application

development Code building
Code coverage
Unit testing
Packaging
Deployment

Q117. How “Infrastructure code” is processed or executed in AWS?

Answer: In AWS, Infrastructure code will be in simple JSON format

After that JSON code will be organized into files called templates

You can Implement the templates on AWS DevOps and then managed as stacks

At last the creating, deleting, updating, etc. operation in the stack are done by Cloud Formation

Q118. Which scripting language is most important for a DevOps engineer?

Answer: It is very important to choose the simplest language for DevOps engineer. Python Language is most suitable language for DevOps.

Q119. How DevOps helps developers?

Answer: Developers can fix bug and implement new features with less time by the help of DevOps. DevOps can also help to build a perfect communication system in a team with every team member.

Q120. Which are popular tools for DevOps?

Answer: Popular tools for DevOps are Jenkins
Nagios
Monit
ELK (Elasticsearch, Logstash, Kibana)
Jenkins
Docker
Ansible
Git

Q121. What is the usefulness of SSH?

Answer: SSH is used to log into a remote machine and work on the command line and also used it to dig into the system to make possible secure coded communications between two untrusted hosts over an insecure network.

Q122. How you would handle revision (version) control?

Answer: I will post the code on SourceForge or GitHub to give avisual for everyone. I will post the checklist also from the last revision to make sure that any unsolved issues are resolved.

Q123. How many types of Http re

Quests are?

Answer: The types of Http re

Quests are
GET
HEAD
PUT
POST
PATCH
DELETE
TRACE
CONNECT
OPTIONS

Q124.If a Linux-build-server suddenly starts getting slow what will you check?

Answer: If a Linux-build-server suddenly starts getting slow, I will check for following three things Application Level troubleshooting: Issues related with RAM, Issues related with Disk I/O read write, Issues related with Disk space, etc. System-Level troubleshooting: Check for Application log file OR application server log file, system performance issues, Web Server Log – check HTTP, tomcat log, etc. or check jboss, WebLogic logs to see if the application server response/receive time is the issues for slowness, Memory Leak of any application Dependent Services troubleshooting: Issues related with Antivirus, Issues related with Firewall, Network issues, SMTP server response time issues, etc

Q125. Describe the key components of DevOps.

The most important DevOps components are: Continuous Integration
Continuous Testing
Continuous Delivery
Continuous Monitoring

Q126. Give example of some popular cloud platform used for DevOps Implementation.

Answer: For DevOps implementation popular Cloud platforms are: Google Cloud
Amazon Web Services
Microsoft Azure

Q127. Describe benefits of using Version Control system.

Answer: Version Control system gives scope to team members to work on any file at suitable time.

All the previous versions and variants are closely packed up inside the VCS.

You can use distributed VCS to store the complete project history in case central server breakdown you can use your team member’s file location storage related with the project.

You can see the actual changes made in the file’s content.

Q128. How Git Bisect helps?

Answer: Git bisect helps you to find the commit which introduced a bug using binary search.

Q129. What is the build?

Answer: Build is a method in which you can put source code together for checking that is the source code working as a single unit. In the build creation process, the source code will undergo compilation, inspection, testing, and deployment.

Q130. What is Puppet?

Answer: Puppet is a project management tool which helps you to convert the administration tasks automatically.

Q131.What is two-factor authentication?

Answer: Two-factor authentication is a security method in which the user provides two ways of identification from separate categories.

Q132. What is ‘Canary Release’?

Answer: It is a pattern which lowers the risk of new version software introduction into the production environment. User will get “Canary Release” in a controlled manner before making it available to the complete user set.

Q133.What are the important types of testing re

Quired to ensure new service is ready for production?

Answer: You need to run continuous testing to make sure the new service is ready for production.

Q134. What is Vagrant?

Answer: Vagrant is a tool used to create and manage a virtual version of computing environments for tests and software development.

Q135. Usefulness of PTR in DNS.

Answer: PTR or Pointer record is used for reverse DNS lookup.

Q136. What is Chef?

Answer: Chef is a powerful automation platform used for transforming infrastructure into code. In this tool, you can use write scripts that are used to automate processes.

Q137. Prere

Quisites for the implementation of DevOps.

Answer: Following are the useful prere

Quisites for DevOps Implementation: At least one Version Control Software (VCS).
Establish communication between the team members
Automated testing
Automated deployment

Q138. For DevOps success which are the best practices?

Answer: Here, are essential best practices for DevOps implementation: The speed of delivery means time taken for any task to get them into the production environment.
Track the defects are found in the various
It’s important to calculate the actual or the average time taken to recover in case of a failure in the production environment.
Get a feedback from the customer about bug report because it also affects the

Quality of application.

Q139. How SubGit tool helps?

Answer: SubGit helps you to move SVN to Git. You can build a writable Git mirror of a local or alien to Subversion repository by using SubGit.

Q140. Name some of the prominent network monitoring tools.

Answer: Some most prominent network monitoring tools are: Splunk
Icinga 2
Wireshark
Nagios
OpenNMS

Q141. How do you know if your video card can run Unity?

Answer: When you use a command

1 /usr/lib/Linux/unity_support_test-p

it will give detailed output about Unity's re

Quirements, and if they are met, then your video card can run unity.

Q142. How to enable startup sound in Ubuntu?

Answer: To enable startup sound Click control gear and then click on Startup Applications

In the Startup Application Preferences window, click Add to add an entry

Then fill the information in comment boxes like Name, Command, and Comment 1 /usr/bin/canberra-gtk-play—id= "desktop-login"—description= "play login sound" Logout and then login once you are done You can use shortcut key Ctrl+Alt+T to open .

Q143. Which is the fastest way to open an Ubuntu terminal in a particular directory?

Answer: To open an Ubuntu terminal in a particular directory, you can use custom keyboard short cut. To do that, in the command field of a new custom keyboard, type genome – terminal – – working – directory = /path/to/dir.

Q144. How could you get the current colour of the current screen on the Ubuntu desktop?

Answer: You have to open the background image in The Gimp (image editor) and use the dropper tool to select the colour on a selected point. It gives you the RGB value of the colour at that point.

Q145. How can you create launchers on a desktop in Ubuntu?

Answer: You have to use ALT+F2 then type" gnome-desktop-item-edit –create-new~/desktop," it will launch the old GUI dialog and create a launcher on your desktop in Ubuntu.

Q146. Explain what Memcached is?

Answer: Memcached is an open source and free, high-performance, distributed memory object caching system. The primary objective of Memcached is to increase the response time for data otherwise it can be recovered or constructed from some other source or database. Memcached is used to reduce the necessity of S

QL database operation or another source repetitively to collect data for a simultaneous re

Quest. Memcached can be used for Social Networking->Profile Caching
Content Aggregation -> HTML/ Page Caching
Ad targeting -> Cookie/profile tracking
Relationship -> Session caching
E-commerce -> Session and HTML caching
Location-based services -> Database

Query scaling
Gaming and entertainment -> Session caching Memcache helps in Make application processes much faster
Memcached make the object selection and rejection process
Reduce the number of retrieval re

Quests to the database
Cuts down the I/O (Input/Output) access (hard disk) Drawback of Memcached is It is not a preserving data store
Not a database
It is not an application specific
Unable to cache large object

Q147. Mention some important features of Memcached?

Answer: Important features of Memcached includes CAS Tokens: A CAS token is attached to an object retrieved from a cache. You can use that token to save your updated object.
Callbacks: It simplifies the code
getDelayed: It decrease the time consumption of your script, waiting for results to come back from a server
Binary protocol: You can use binary protocol instead of ASCII with the newer client
Igbinary: A client always has to do serialization of the value with complex data previously, but now with Memcached, you can use igbinary option.

Q148. Is it possible to share a single instance of a Memcache between multiple projects?

Answer: Yes, it is possible to share a single instance of Memcache between multiple projects. You can run Memcache on more than one server because it is a memory store space. You can also configure your client to speak to a particular set of case. So, you can run two different Memcache processes on the same host independently.

Q149. You are having multiple Memcache servers, one of the memcache servers fails, and it has your data, can you recover key data from the perticular failed server?

Answer: Data won’t be removed from the server but there is a solution for auto-failure, which you can configure for multiple nodes. Fail-over can be triggered during any socket or Memcached server level errors and not during standard client errors like adding an existing key, etc.

Q150. How can you minimize the Memcached server outages?

Answer: If you write the code to minimize cache stampedes then it will leave a minimal impact
Another way is to bring up an instance of Memcached on a new machine using the lost machines IP address
The code is another option to minimize server outages as it gives you the liberty to change the Memcached server list with minimal work
Setting timeout value is another option that some Memcached clients implement for Memcached server outage. When your Memcached server goes down, the client will keep trying to send a re

Quest till the time-out limit is reached

Q151. How can you update Memcached when data changes?

Answer: When data changes you can update Memcached by Clearing the Cache proactively: Clearing the cache when an insert or update is made

Resetting the Cache: this method is similar with previous one but without delete the keys and wait for the next re

Quest for the data to refresh the cache, reset the values after the insert or update.

Q152. What is Dogpile effect? What is the prevention of this effect?

Answer: When a cache expires, and websites are hit by the multiple re

Quests made by the client at the same time the Dogpile effect occurs. You have to use semaphore lock to prevent the effect. In this system after value expires, the first process ac

Quires the lock and starts generating new value.

Q153. How Memcached should not be used?

Answer: You have to use Memcached as cache; don't use it as a data store.
Don't use Memcached as the ultimate source of information to run your application. You must always have an option of data source in your hand.
Memcached is basically a value store and can't perform a

Query over the data or go through again over the contents to extract information.
Memcached is not secure either in encryption or authentication.

Q154. When a server gets shut down does data stored in Memcached is still available?

Answer: No after a server shuts down and then restart the stored data in Memcached will be deleted because Memcached is unable to store data for long time.

Q155. What are the difference between Memcache and Memcached?

Answer: Memcache: It is an extension that allows you to work through handy object-oriented (OOP's) and procedural interfaces. It is designed to reduce database load in dynamic web applications.
Memcached: It is an extension that uses the libmemcached library to provide API for communicating with Memcached servers. It is used to increase the dynamic web applications by reducing database load. It is the latest API.

Q156. Explain Blue/Green Deployment Pattern

Answer: Blue/Green colouring pattern is one of the hardest challenge faced at the time of automatic deployment process. In Blue/ Green Deployment approach, you need to make sure two identical production environments. Only one among them is LIVE at any given point of time and it is called Blue environment. After take the full preparation to release the software the team conducts the final testing in an environment called Green environment. When the verification is complete the traffic is routed to the Green environment.

Q157. What are the containers?

Answer: Containers are from of lightweight virtualization and create separation among process.

Q158. What is post mortem meeting with reference to DevOps?

Answer: In DevOps Post mortem meeting takes place to discuss about the mistakes and how to repair the mistakes during the total process.

Q159. What is the easiest method to build a small cloud?

Answer: VMfres is one of the best options to built IaaS cloud from Virtual Box VMs in lesser time. But if you want lightweight PaaS, then Dokku is a better option because bash script can be PaaS out of Dokku containers.

Q160. Name two tools you can use for docker networking.

Answer: You can use Kubernetes and Docker swarm tools for docker networking.

Q161. Name some of DevOps Implementation area

Answer: DevOps are used for Production, Production feedback, IT operation, and its software development.

Q162. What is CBD'?

Answer: CBD or Component-Based Development is a uni

Que way to approach product development. In this method, Developers don’t develop a product from scratch, they look for existing well defined, tested, and verified components to compose and assemble them to a product.

Q163. Explain Pair Programming with reference to DevOps

Answer: Pair programming is an engineering practice of Extreme Programming Rules. This is the process where two programmers work on the same system on the same design/algorithm/code. They play two different roles in the system. One as a“driver” and other as an “observer”. Observer continuously observes the progress of a project to identify problems. T

hey both can change their roles in a step of the program.

Q1). Describe what DevOps is?

DevOps is the new buzz in the IT world, swiftly spreading all through the technical space. Like other new and popular technologies, people have contradictory impressions of what DevOps is exactly. The main objective of DevOps is to alter and improve the relationship between the development and IT teams by advocating better inter-communication and smoother collaboration between two units of an enterprise.

Q2). What is the programming language used in DevOps?

Python is used in DevOps.

Q3). What is the necessity of DevOps?

Corporations are now facing the necessity of carrying quicker and improved requests to see the ever more persistent demands of mindful users to decrease the “Time to Marketplace.“ DevOps often benefits placement to occur very profligately.

Q4). Which are the areas where DevOps is implemented?

By the passage of time, the need for DevOps is continuously increasing. However, these are the main areas it is implemented in-

Areas of Production Development areas

production feedback

development of IT Operations

Q5). What is agile expansion and Scrum?

Agile growth used as a substitute for Waterfall development training. In Agile, the expansion process is more iterative and additive; there are more challenging and response at every stage of development as opposed to only the latter stage in Waterfall. Scrum is used to accomplish composite software and product growth, using iterative and additive performs. Scrum has three roles:

Product owner

Scrum master

Team

Q6). Name a few most famous DevOps tools?

The most prevalent DevOps tools are stated below:

Puppet

Chef

Ansible

Git

Nagios

Docker

Jenkins

Q7). Can we consider DevOps as an agile practice?

Yes, DevOps is considered as an agile practice where development is driven by profound changing demands of professionals to stick closer to the corporate needs and requirements

Q8). What is DevOps engineer's responsibility concerning Agile development?

DevOps specialist exertion very methodically with Agile development teams to assurance they have a condition essential to support purposes such as automatic testing, incessant Integration, and unceasing Delivery. DevOps specialist must be in continuous contact with the developers and make all compulsory parts of environment work flawlessly.

Q9). Why is Incessant Testing significant for DevOps?

You can respond to this question by saying, "Incessant Testing permits any change made in the code to be tested directly. This circumvents the glitches shaped by having "big-bang" testing left-hand to the end of the series such as announcement postponements and quality matters. In this way, Incessant Testing eases more recurrent and good class releases."

Q10). What do you think is the role of SSH?

SSH is a Secure Shell which gives the users a very secure as well encrypted mechanism to safely log into systems and ensures the safe transfer of files. It aids in the process of logging out of a remote machine along with the work on the command line. It helps in securing an encrypted and protected end to end communications between two hosts communicating over an insecure network.

Q11). How will you differentiate DevOps from Agile?

Agile is the technology which is all about software development, whereas DevOps is the technology used for software deployment and management.

Q12). What are the benefits of DevOps when seen from the Technical and Business viewpoint?

The Technical assistance features of DevOps can be given as:

Software delivery is incessant.

Decreases Difficulty in problems.

Quicker approach to resolve problems

Workforce is abridged.

Business welfare features:

A high degree of bringing its features

Steady operating environments

More time increased to Add values.

Allowing quicker feature time to market

Q13). Why do you think DevOps is developers friendly?

DevOps is developers friendly because it fixes the bugs and implements the new features very smoothly quickly. It is amazing because it provides the much-needed clarity of communication among team members.

Q14). What measures would you take to handle revision (version) control?

To manage a successful revision control, you are required to post your code on SourceForge or GitHub so that everyone on the team can view it from there and also there is an option for viewers to give suggestions for the better improvement of it.

Q15). List a few types of HTTP requests.

A few types of Http requests are”

GET

HEAD

PUT

POST

PATCH

DELETE

TRACE

CONNECT

OPTIONS

Q16). Explain the DevOps Toolchain.

Here is the DevOps toolchain-

Code

Build

Test

Package

Release

Configure

Monitor

Q17). Elucidate the core operations of DevOps concerning development and Infrastructure.

Here is a list of the core operations of DevOps:

Unit testing

Packaging

Code coverage

Code developing

Configuration

Orchestration

Provisioning

Deployment

Q18). Why do you think there is a need for Continuous Integration of Development & Testing?

Continuous Integration of Development and Testing enhances the quality of software and highly deducts the time which is taken to deliver it, by replacing the old-school practice of testing only after completing all the development process.

Q19). Name a few branching strategies used in DevOps

A few branching strategies to be used are-

Feature Branching

Task Branching

Release Branching

Q20). What is the motive of GIT tools in DevOps?

Read: What is the Difference between Agile and DevOps

The primary objective of Git is to efficaciously manage a project or a given bundle of files as they keep on changing over time. Git tool stores this important information in a data structure kind of thing called a Git repository.

Q21). Explain what the major components of DevOps are?

The major components of DevOps are continuous integration, continuous delivery, continuous integration, and continuous monitoring.

Q22). What steps should be taken when Linux-based-server suddenly gets slow?

When a Linux-based-server suddenly becomes slow, then you should focus on three things primarily:

Application level troubleshooting

System level troubleshooting

Dependent level troubleshooting

Q23). Which cloud platforms can be used for the successful DevOps implementation?

Cloud platforms that can be used for the successful DevOps implementation are given as:

Google Cloud

Amazon Web Services

Microsoft Azure

Q24). What is a Version Control System (VCS)?

VCS is a software application that helps software developers to work together and maintain the complete history of their work.

Q25). What are the significant benefits of VCS (Version Control System)?

The significant benefits of using VCS can be given as:

It allows team members to work simultaneously.

All past variants and versions are packed within VCS.

A distributed VCS helps you to store the complete history of the project. In case of a breakdown of the central server, you may use the local GIT repository.

It allows you to see what exact changes are made to the content of a file.

Q26). What is a Git Bisect?

Git Bisect helps you to find the commit which introduced a bug using the binary search. Here is the basic syntax for a Git Bisect: Git bisect

Q27). What do you understand by the term build?

A build is a method in the source code where the source code is put together to check how it works as a single unit. In the complete process, the source code will undergo compilation, testing, inspection, and deployment.

Q28). As per your experience, what is the most important thing that DevOps helps to achieve?

The most important thing that DevOps helps us to achieve is to get the changes in a product quickly while minimizing risks related to software quality and compliance. Other than this, there are more benefits of DevOps that include better communication, better collaboration among team members, etc.

Q29). Discuss one use case where DevOps can be implemented in the real-life.

Etsy is a Company that focuses on vintage, handmade, and uniquely manufactured items. There are millions of Etsy users who are selling products online. At this stage, Etsy decided to follow a more agile approach. DevOps helped Etsy with a continuous delivery pipeline and fully automated deployment lifecycle.

Q30). Explain your understanding of both the software development side and technical operations side of an organization you have worked in the past recently.

The answer to this question may vary from person to person. Here, you should discuss the experience of how flexible you were in your last Company.

DevOps Interview Questions and Answers for advanced workforce

In this section, we will be discussing interview questions for experienced people having more than three years of experience. Before you go through questions directly, take this quiz first to become a little more confident in your skills.

Q31). What are the anti-patterns in DevOps?

A pattern is used by others, not by organizations and you continue blindly follow it. You are essentially adopting anti-patterns here.

Q32). What is a Git Repository?

It is a version control system that tracks changes to a file and allows you to revert to any particular changes.

Q33). In Git, how to revert a commit that has already been made public?

Remove or fix the commit and push it to the remote repository. This is the most natural style to fix an error. To do this, you should use the command given below: Git commit –m “commit message”

Create a new commit that undergoes all changes that were made in the bad commit. Git revert

Q34). What is the process to squash last N number of commits into a single commit?

There are two options to squash last N number of commits into a single commit.

To write a new commit message from scratch, you should use the following command: git reset –soft HEAD ~N && git commit

To edit the existing message, you should extract those messages first and pass them to the Git commit for later usage. Git reset –soft HEAD ~ N&& git commit –edit –m “$(git log –format=%B –reverse .HEAD {N})”

Q35). What is Git rebase and how to use it for resolving conflicts in a feature branch before merging?

Git Rebase is a command that is used to merge another branch to the existing branch where you are working recently. It moves all local commits at the top of the history of that branch. It effectively replays the changes of feature branch at the tip of master and allowing conflicts to be resolved in the process. Moreover, the feature branch can be merged to the master branch with relative ease and sometimes considered as the fast-forward operation.

Q36). How can you configure a git repository to run code sanity checking tools right before making commits and preventing them if the test fails?

Sanity or smoke test determines how to continue the testing reasonably. This is easy configuring a Git repository to run code sanity checking before making commits and preventing them if the test fails. It can be done with a simple script as mentioned below:

```
#!/bin/sh

file=$(git diff -cached -name-only -diff-filter=ACM | grep '.go$')

if [ -z file]; then exit 0

fi

unfmtd=$(gofmt -l $files)

if [ -z unfmtd]; then

exit 0

fi

eacho "some .go files are not fmt'd"

exit 1
```

Q37). How to find a list of files that are changed in a certain manner?

To get a list of files that are changed or modified in a particular way, you can use the following command: git diff-tree -r{hash}

Q38). How to set up a script every time a repository receives new commits from a push?

There are three techniques to set up a script every time a repository receives new commits from Push. These are the pre-receive hook, post-receive hook, and update hook, etc.

Q39). Write commands to know in Git if a branch is merged to the master or not.

Here are the commands to know in Git if a branch is merged to the master or not. To list branches that are merged to the current branch, you can use the following command: git branch -merged

To list branches that are not merged to the current branch, you can use the following command: git branch –no-merged

Q40). What is continuous integration in DevOps?

It is a development practice that requires developers to integrate code into a shared repository multiple times a day. Each check-in is verified with an automated build allowing teams to detect problems early.

Q41). Why is continuous integration necessary for the development and testing team?

It improves the quality of software and reduces the overall time to product delivery, once the development is complete. It allows the development team to find and locate bugs at an early stage and merge them to the shared repository multiple times a day for automating testing.

Q42). Are there any particular factors included in continuous integration?

These following points you should include to answer this question:

Automate the build and maintain a code repository.

Make the build self-tested and fast.

Testing should be done in a clone of the production environment.

It is easy getting the latest deliverables.

Automate the deployment, and everyone should be able to check the result of the latest build.

Q43). What is the process to copy Jenkins from one server to another?

There are multiple ways to copy Jenkins from one server to another. Let us discuss them below:

You can move the job from one Jenkin installation to another by simply copying the corresponding job directory.

Make a copy of the existing job and save it with a different name in the job directory.

Rename the existing job and make necessary changes as per the requirement.

Q44). How to create a file and take backups in Jenkins?

For taking backup in Jenkins, you just need to copy the directory and save it with a different name.

Q45). Explain the process to set up jobs in Jenkins.

Go to the Jenkins page at the top, select the “new job” option, and choose “Build a free-style software project.”

Select the optional SCM where your source code resides.

Select the optional triggers to control when Jenkins performs builds.

Choose the preferable script that can be used to make the build.

Collect the information for the build and notify people about the build results.

Q46). Name a few useful plugins in Jenkins.

Some popular plugins in Jenkins can be given as:

Read: What is Git? Git Tutorial Guide for Beginners

Maven 2 project

Amazon EC2

HTML publisher

Copy artifact

Join

Green Balls

Q47). How will you secure Jenkins?

Here are a few steps you should follow to secure the Jenkins:

Make sure that global security option is on and Jenkins is integrated with the company’s user directory with appropriate login details.

Make sure that the project matrix is enabled for the fine tune access.

Automate the process of setting privileges in Jenkins with custom version-controlled scripts.

Limit the physical access to Jenkins data/folders.

Run the security audits periodically.

Jenkins is one of the popular tools used extensively in DevOps and hands-on training in Jenkins can make you an expert in the DevOps domain.

Q48). What is continuous testing in DevOps?

It is the process of executing automated tests as part of software delivery to receive immediate feedback within the latest build. In this way, each build can be tested continuously allowing the development team to get faster feedback and avoid potential problems from progressing to the next stage of the delivery cycle.

Q49). What is automation testing in DevOps?

It is the process of automating the manual process for testing an application under test (AUT). It involves the usage of different testing tools that lets you creating test scripts that can be executed repeatedly and does not require any manual intervention.

Q50). Why is automation testing significant in DevOps?

The automation testing is significant for the following reasons in DevOps:

It supports the execution of repeated test cases.

It helps in testing a large test matrix quickly.

It helps in enabling the test execution.

It encourages parallel execution.

It improves accuracy by eliminating human intervened errors.

It helps in saving the overall time and investments.

Q51). What is the importance of continuous testing in DevOps?

With continuous testing, all changes to the code can be tested automatically. It avoids the problem created by the big-bang approach at the end of the cycle like release delays or quality issues etc. In this way, continuous testing assures frequent and quality releases.

Q52). What are the major benefits of continuous testing tools?

The major benefits of continuous testing tools can be given below.

Policy analysis

Risk assessment

Requirements traceability

Test optimization

Advanced analytics

Service virtualization

Q53). Which testing tool is just the best as per your experience?

Selenium testing tool is just the best as per my experience. Here are a few benefits which makes it suitable for the workplace.

It is an open source free testing tool with a large user base and helping communities.

It is compatible with multiple browsers and operating systems.

It supports multiple programming languages with regular development and distributed testing.

Q54). What are the different testing types supported by the Selenium?

These are the Regression Testing and functional testing.

Q55). What is two-factor authentication in DevOps?

Two-factor authentication in DevOps is a security method where the user is provided with two identification methods from different categories.

Q56). Which type of testing should be performed to make sure that a new service is ready for production?

It is continuous testing that makes sure that a new service is ready for production.

Q57). What is Puppet?

It is a configuration management tool in DevOps that helps you in automating administration tasks.

Q58). What do you understand by the term Canary Release?

It is a pattern that reduces the risk of introducing a new version of the software into the production environment. It is made available in a controlled manner to the subset of users before releasing to the complete set of users.

Q59). What is the objective of using PTR in DNS?

PTR means pointer record that is required for a reverse DNS lookup.

Q60). What is Vagrant in DevOps?

It is a DevOps tool that is used for creating and managing virtual environments for testing and developing software programs.

DevOps Job Interview Questions and Answers

Q61). What are the prerequisites for the successful implementation of DevOps?

Here are the prerequisites for the successful implementation of DevOps:

One Version control system

Automated testing

Automated deployment

Proper communication among team members

Q62). What are the best practices to follow for DevOps success?

Here are the essential practices to follow for DevOps success:

The speed of delivery time taken for a task to get them into the production environment.

Focus on different types of defects in the build.

Check the average time taken to recover in case of failure.

The total number of reported bugs by customers impacting the quality of an application.

Q63). What is a SubGit tool?

A SubGit tool helps in migrating from SVN to Git. It allows you to build a writable Git mirror of a remote or local subversion repository.

Q64). Name a few networks migrating tools.

Splunk

Icinga 2

Wireshark

Nagios

OpenNMS

Q65). How to check either your video card can run Unity or not?

Here is the command to check either your video card can run unity or not: /usr/lib/linux/unity_support_test-p

It will give you a depth of unity's requirements. If they are met, your video card can run Unity.

Q66). How to enable the start-up sounds in ubuntu?

To enable the start-up sounds in Ubuntu, you should follow these steps:

Click control gear then click on startup applications.

In the "startup application preferences" window, click "Add" to add a new entry.

Add the following command in the comment boxes: /usr/bin/Canberra-gtk-play-id= "desktop-login" –description= "play login sound"

Now, log out from the account once you are done.

Q67). What is the quickest way of opening an Ubuntu terminal in a particular directory?

For this purpose, you can use the custom keyword shortcuts.

To do that, in the command field of a new custom keyboard, type genome –terminal –working –directory = /path/to/dir.

Q68). How to get the current color of the screen on the Ubuntu desktop?

You should open the background image and use a dropper tool to select the color at a specific point. It will give you the RGB value for that color at a specific point.

Q69). How to create launchers on a Ubuntu Desktop?

To create a launcher on a Ubuntu desktop, you should use the following command:

ALT+F2 then type “gnome-desktop-item-edit-create-new~/desktop,” it will launch the old GUI dialog and create a launcher on your desktop

Q70). What is Memcached in DevOps?

It is an open source, high speed, distributed memory object. Its primary objective is enhancing the response time of data that can otherwise be constructed or recovered from another source of database. It avoids the need for operating SQL database repetitively to fetch data for a concurrent request.

DevOps quiz

Q71). Why Memcached in useful?

It speeds up the application processes.

It determines what to store and share.

It reduces the total number of retrieval requests to the database.

It cuts the I/O access from the hard disk.

Q72). What are the drawbacks of Memcached?

It is not a persistent data store

It is not a database.

It is not application-specific.

It is not able to cache large objects.

Q73). What are the features of Memcached?

A few highlighted features of Memcached can be given as:

CAS Tokens that are used to store the updated objects.

Callbacks to simplify the code.

GetDelayed to reduce the response or wait time for the outcome.

A binary protocol to use with the new client.

Igbinary data option is available to use with the complex data.

Q74). Can you share a single instance of Memcached with multiple instances?

Read: Top 20 Git Interview Questions and Answers 2018

Yes, it is possible.

Q75). If you have multiple Memcached servers and one of the Memcached servers gets failed, then what will happen?

Even if one of the Memcached servers gets failed, data won't get lost, but it can be recovered by configuring it for multiple nodes.

Q76). How to minimize the Memcached server outages?

If one of the server instances get failed, it will put a huge load on the database server. To avoid this, the code should be written in such a way that it can minimize the cache stampedes and leave a minimal impact on the database server.

You can bring up an instance of Memcached on a new machine with the help of lost IP addresses.

You can modify the Memcached server list to minimize the server outages.

Set up the timeout value for Memcached server outages. If the server gets down, it will try to send a request to the client until the timeout value is achieved.

Q77). How to update Memcached when data changes?

To update the Memcached in case of data changes, you can use these two techniques:

Clear the cache proactively

Reset the Cache

Q78). What is a Dogpile effect and how to prevent it?

Dogpile effect refers to the event when the cache expires, and website hits by multiple requests together at the same time. The semaphore lock can minimize this effect. When the cache expires, the first process acquires the lock and generates new value as required.

Q79). Explain when Memcached should not be used?

It should not be used as a datastore but a cache only.

It should not be taken the only source of information to run your apps, but the data should be available through other sources too.

It is just a value store or a key and cannot perform a query or iterate over contents to extract the information.

It does not offer any security for authentication or encryption.

Q80). What is the significance of the blue/green color in deployment pattern?

These two colors are used to represent tough deployment challenges for a software project. The live environment is the Blue environment. When the team prepares the next release of the software, it conducts the final stage of testing in the Green environment.

Q81). What is a Container?

Containers are lightweight virtualizations that offer isolation among processes.

Q82). What is post mortem meeting in DevOps?

A post mortem meeting discusses what went wrong and what steps to be taken to avoid failures.

Q83). Name two tools that can be used for Docket networking.

These are Docker Swarm and Kubernetes.

Q84). How to build a small cloud quickly?

Dokku can be a good option to build a small cloud quickly.

Q85). Name a few common areas where DevOps is implemented?

These are IT, production, operations, marketing, software development, etc.

Q86). What is pair programming in DevOps?

It is a development practice of extreme programming rules.

Q87). What is CBD in DevOps?

CBD or component-based development is a unique style of approaching product development.

Q88). What is Resilience Test in DevOps?

It ensures the full recovery of data in case of failure.

Q89). Name a few important DevOps KPIs.

Three most important KPIs of DevOps can be given as:

Meantime to failure recovery

Percentage of failed deployments

Deployment Frequency

Q90). What is the difference between asset and configuration management?

Asset management refers to any system that monitors and maintains things of a group or unit. Configuration Management is the process of identifying, controlling, and managing configuration items in support of change management.

Q91). How does HTTP work?

An HTTP protocol works like any other protocol in a client-server architecture. The client initiates a request, and the server responds to it.

Q92). What is Chef?

It is a powerful automated tool for transforming infrastructure into code.

Q93). How will you define a resource in Chef?

A resource is a piece of infrastructure and its desires state like packages should be installed, services should be in running state, the file could be generated, etc.

Q94). How will you define a recipe in Chef?

A recipe is a collection of resources describing a particular configuration or policy.

Q95). How is cookbook different from the recipe in Chef?

The answer is pretty direct. A recipe is a collection of resources, and a Cookbook is a collection of recipes and other information.

Q96). What is an Ansible Module?

Modules are considered as a unit of work in Ansible. Each module is standalone, and it can be written in common scripting languages.

Q97). What are playbooks in Ansible?

Playbooks are Ansible’s orchestration, configuration, and deployment languages. They are written in human-readable basic text language.

Q98). How can you check the complete list of Ansible variables?

You can use this command to check the complete list of Ansible variables. Ansible –m setup hostname

Q99). What is Nagios?

It is a DevOps tool for continuous monitoring of systems, business processes, or application services, etc.

Q100). What are plugins in DevOps?

Plugins are scripts that are run from a command line to check the status of Host or Service.

Question: **What Are Benefits Of DevOps?**

DevOps is gaining more popularity day by day. Here are some benefits of implementing DevOps Practice.

Release Velocity: DevOps enable organizations to achieve a great release velocity. We can release code to production more often and without any hectic problems.

Development Cycle: DevOps shortens the development cycle from initial design to production.

Full Automation: DevOps helps to achieve full automation from testing, to build, release and deployment.

Deployment Rollback: In DevOps, we plan for any failure in deployment rollback due to a bug in code or issue in production. This gives confidence in releasing feature without worrying about downtime for rollback.

Defect Detection: With DevOps approach, we can catch defects much earlier than releasing to production. It improves the quality of the software.

Collaboration: With DevOps, collaboration between development and operations professionals increases.

Performance-oriented: With DevOps, organization follows performance-oriented culture in

which teams become more productive and more innovative.

Question: What Is The Typical DevOps workflow?

The typical DevOps workflow is as follows:

- Atlassian Jira for writing requirements and tracking tasks.
- Based on the Jira tasks, developers checking code into GIT version control system.
- The code checked into GIT is built by using Apache Maven.
- The build process is automated with Jenkins.
- During the build process, automated tests run to validate the code checked in by a developer.
- Code built on Jenkins is sent to organization's Artifactory.
- Jenkins automatically picks the libraries from Artifactory and deploys it to Production.
- During Production deployment, Docker images are used to deploy same code on multiple hosts.
- Once a code is deployed to Production, we use monitoring tools like ngios are used to check the health of production servers.
- Splunk based alerts inform the admins of any issues or exceptions in production.

Question: DevOps Vs Agile?

Agile is a set of values and principles about how to develop software in a systematic way.

Where as DevOPs is a way to quickly, easily and repeatably move that software into production infrastructure, in a safe and simple way.

In oder to achieve that we use a set of DevOps tools and techniques.

Question: **What Is The Most Important Thing DevOps Helps Us To Achieve?**

Most important aspect of DevOps is to get the changes into production as quickly as possible while minimizing risks in software quality assurance and compliance. This is the primary objective of DevOps.

Question: What Are Some DevOps tools.

Here is a list of some most important DevOps tools

- Git
- Jenkins, Bamboo
- Selenium

- Puppet, BitBucket
- Chef
- Ansible, Artifactory
- Nagios
- Docker
- Monit
- ELK –Elasticsearch, Logstash, Kibana
- Collectd/Collect

Question: **How To Deploy Software?**

Code is deployed by adopting continuous delivery best practices. Which means that checked in code is built automatically and then artifacts are published to repository servers.

On the application severs there are deployment triggers usually timed by using cron jobs.

All the artifacts are then downloaded and deployed automatically.

Gradle DevOps Interview Questions

Question: **What is Gradle?**

Gradle is an open-source build automation system that builds upon the concepts of Apache Ant and Apache Maven. Gradle has a proper programming language instead of XML configuration file and the language is called ‘Groovy’.

Gradle uses a directed acyclic graph ("DAG") to determine the order in which tasks can be run.

Gradle was designed for multi-project builds, which can grow to be quite large. It supports incremental builds by intelligently determining which parts of the build tree are up to date, any task dependent only on those parts does not need to be re-executed.

Question: What Are Advantages of Gradle?

Gradle provides many advantages and here is a list

- **Declarative Builds:** Probably one of the biggest advantage of Gradle is Groovy language. Gradle provides declarative language elements. Which providea build-by-convention support for Java, Groovy, Web and Scala.
- **Structured Build:** Gradle allows developers to apply common design principles to their build. It provides a perfect structure for build, so that well-structured and easily maintained, comprehensible build structures can be built.
- **Deep API:** Using this API, developers can monitor and customize its configuration and execution behaviors.
- **Scalability:** Gradle can easily increase productivity, from simple and single project builds to huge enterprise multi-project builds. **Multi-project builds:** Gradle
- supports multi-project builds and also partial builds.
- **Build management:** Gradle supports different strategies to manage project dependencies.
- **First build integration tool** − Gradle completely supports ANT tasks, Maven and lvy repository infrastructure for publishing and retrieving dependencies. It also provides a converter for turning a Maven pom.xml to Gradle script.
- **Ease of migration:** Gradle can easily adapt to any project structure.

- **Gradle Wrapper:** Gradle Wrapper allows developers to execute Gradle builds on machines where Gradle is not installed. This is useful for continuous integration of servers.

- **Free open source** − Gradle is an open source project, and licensed under the Apache Software License (ASL).

- **Groovy:** Gradle's build scripts are written in Groovy, not XML. But unlike other approaches this is not for simply exposing the raw scripting power of a dynamic language. The whole design of Gradle is oriented towards being used as a language, not as a rigid framework.

Question: Why Gradle Is Preferred Over Maven or Ant?

There isn't a great support for multi-project builds in Ant and Maven. Developers end up doing a lot of coding to support multi-project builds.

Also having some build-by-convention is nice and makes build scripts more concise. With Maven, it takes build by convention too far, and customizing your build process becomes a hack.

Maven also promotes every project publishing an artifact. Maven does not support subprojects to be built and versioned together.

But with Gradle developers can have the flexibility of Ant and build by convention of Maven.

Groovy is easier and clean to code than XML. In Gradle, developers can define dependencies between projects on the local file system without the need to publish artifacts to repository.

Question: Gradle Vs Maven

The following is a summary of the major differences between Gradle and Apache Maven:

Flexibility: Google chose Gradle as the official build tool for Android; not because build scripts are code, but because Gradle is modeled in a way that is extensible in the most fundamental ways.

Both Gradle and Maven provide convention over configuration. However, Maven provides a very rigid model that makes customization tedious and sometimes impossible.

While this can make it easier to understand any given Maven build, it also makes it unsuitable for many automation problems. Gradle, on the other hand, is built with an empowered and responsible user in mind.

Performance

Both Gradle and Maven employ some form of parallel project building and parallel dependency resolution. The biggest differences are Gradle's mechanisms for work avoidance and incrementally. Following features make Gradle much faster than Maven:

- **Incrementally:**Gradle avoids work by tracking input and output of tasks and only running what is necessary.
- **Build Cache:**Reuses the build outputs of any other Gradle build with the same inputs.
- **Gradle Daemon:**A long-lived process that keeps build information "hot" in memory.

User Experience

Maven's has a very good support for various IDE's. Gradle's IDE support continues to improve quickly but is not great as of Maven.

Although IDEs are important, a large number of users prefer to execute build operations through a command-line interface. Gradle provides a modern CLI that has discoverability features like `gradle tasks`, as well as improved logging and command-line completion.

Dependency Management

Both build systems provide built-in capability to resolve dependencies from configurable repositories. Both are able to cache dependencies locally and download them in parallel.

As a library consumer, Maven allows one to override a dependency, but only by version. Gradle provides customizable dependency selection and substitution rules that can be declared once and handle unwanted dependencies project-wide. This substitution mechanism enables Gradle to build multiple source projects together to create composite builds.

Maven has few, built-in dependency scopes, which forces awkward module architectures in common scenarios like using test fixtures or code generation. There is no separation between unit and integration tests, for example. Gradle allows custom dependency scopes, which provides better-modeled and faster builds.

Question: What are Gradle Build Scripts?

Gradle builds a script file for handling projects and tasks. Every Gradle build represents one or more projects.

A project represents a library JAR or a web application.

Question: What is Gradle Wrapper?

The wrapper is a batch script on Windows, and a shell script for other operating systems.

Gradle Wrapper is the preferred way of starting a Gradle build.

When a Gradle build is started via the wrapper, Gradle will automatically download and run the build.

Question: What is Gradle Build Script File Name?

This type of name is written in the format that is build.gradle. It generally configures the Gradle scripting language.

Question: How To Add Dependencies In Gradle?

In order to make sure that dependency for your project is added, you need to mention the

configuration dependency like compiling the block dependencies of the build.gradle file.

Question: What Is Dependency Configuration?

Dependency configuration comprises of the external dependency, which you need to install well and make sure the downloading is done from the web. There are some key features of this configuration which are:

1. **Compilation:** The project which you would be starting and working on the first needs to be well compiled and ensure that it is maintained in the good condition.
2. **Runtime:** It is the desired time which is required to get the work dependency in the form of collection.
3. **Test Compile:** The dependencies check source requires the collection to be made for running the project.
4. **Test runtime:** This is the final process which needs the checking to be done for running the test that is in a default manner considered to be the mode of runtime

Question: What Is Gradle Daemon?

A daemon is a computer program that runs as a background process, rather than being under the direct control of an interactive user.

Gradle runs on the Java Virtual Machine (JVM) and uses several supporting libraries that require a non-trivial initialization time.

As a result, it can sometimes seem a little slow to start. The solution to this problem is the Gradle *Daemon*: a long-lived background process that executes your builds much more quickly than would otherwise be the case.

We accomplish this by avoiding the expensive bootstrapping process as well as leveraging caching, by keeping data about your project in memory. Running Gradle builds with the Daemon is no different than without

Question: What Is Dependency Management in Gradle?

Software projects rarely work in isolation. In most cases, a project relies on reusable functionality in the form of libraries or is broken up into individual components to compose a modularized system.

Dependency management is a technique for declaring, resolving and using dependencies required by the project in an automated fashion.

Gradle has built-in support for dependency management and lives up the task of fulfilling typical scenarios encountered in modern software projects.

Question: What Are Benefits Of Daemon in Gradle 3.0

Here are some of the benefits of Gradle daemon

1. It has good UX
2. It is very powerful
3. It is aware of the resource
4. It is well integrated with the Gradle Build scans
5. It has been default enabled

Question: What Is Gradle Multi-Project Build?

Multi-project builds helps with modularization. It allows a person to concentrate on one area of work in a larger project, while Gradle takes care of dependencies from other parts of the project

A multi-project build in Gradle consists of one root project, and one or more subprojects that may also have subprojects.

While each subproject could configure itself in complete isolation of the other subprojects, it is common that subprojects share common traits.

It is then usually preferable to share configurations among projects, so the same configuration affects several subprojects.

Question: What Is Gradle Build Task?

Gradle Build Tasks is made up of one or more projects and a project represents what is been done with Gradle.

Some key of features of Gradle Build Tasks are:

1. Task has life cycled methods [do first, do last]
2. Build Scripts are code
3. Default tasks like run, clean etc
4. Task dependencies can be defined using properties like dependsOn

---

Question: What is Gradle Build Life Cycle?

Gradle Build life cycle consists of following three steps

-Initialization phase: In this phase the project layer or objects are organized

-Configuration phase: In this phase all the tasks are available for the current build and a dependency graph is created

-Execution phase: In this phase tasks are executed.

Question: What is Gradle Java Plugin?

The Java plugin adds Java compilation along with testing and bundling capabilities to the project. It is introduced in the way of a SourceSet which act as a group of source files complied and executed together.

Question: What is Dependency Configuration?

A set of dependencies is termed as dependency configuration, which contains some external dependencies for download and installation.

Here are some key features of dependency configuration are:

Compile:

The project must be able to compile together

Runtime:

It is the required time needed to get the dependency work in the collection.

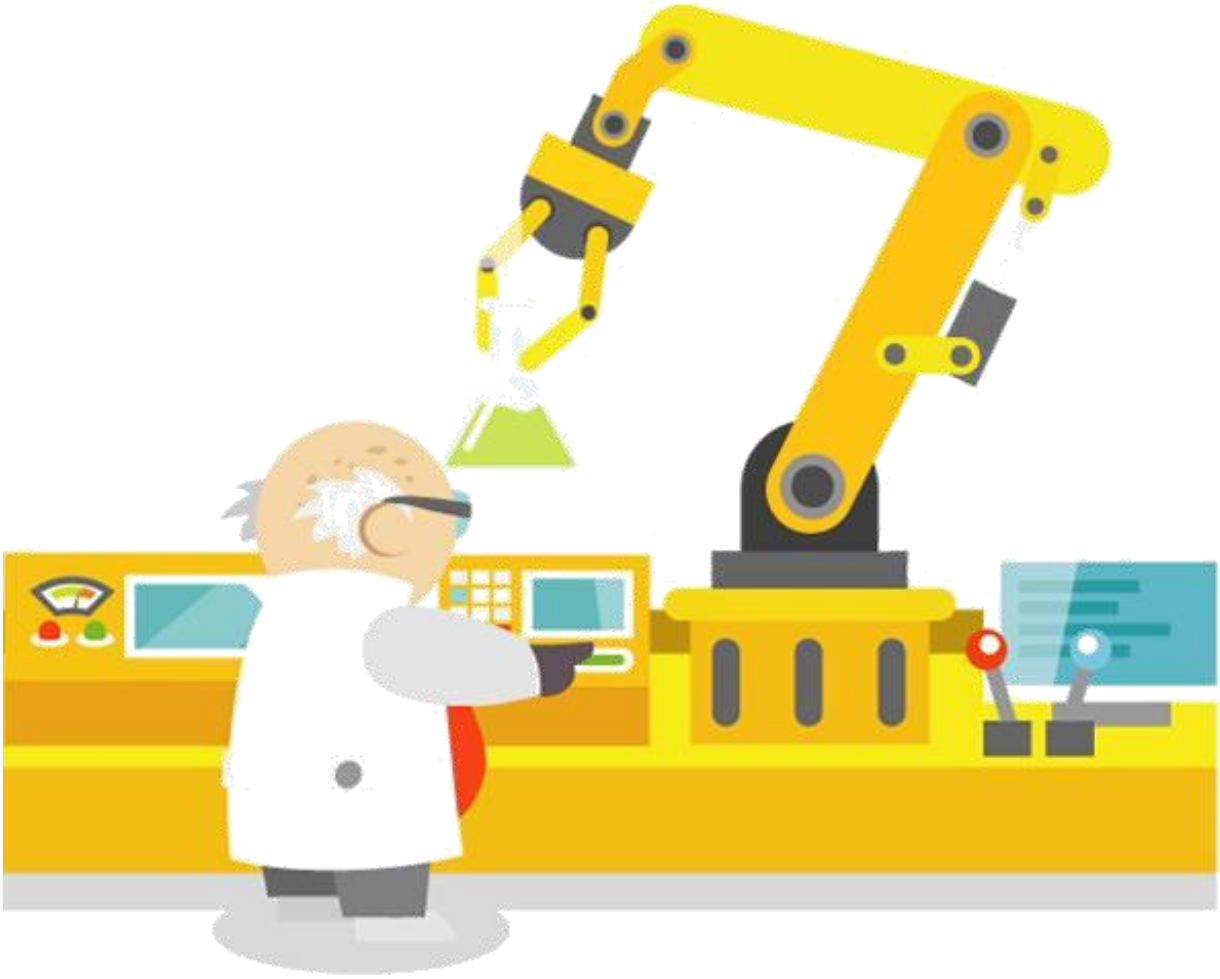
Test Compile:

The check source of the dependencies is to be collected in order to run the project.

Test Runtime:

The final procedure is to check and run the test which is by default act as a runtime mode.

Groovy DevOps Interview Questions



Question: What is Groovy?

Apache Groovy is a object-oriented programming language for the Java platform.

It is both a static and dynamic language with features similar to those of Python, Ruby, Perl, and Smalltalk.

It can be used as both a programming language and a scripting language for the Java Platform, is compiled to Java virtual machine (JVM) bytecode, and interoperates seamlessly with other Java code and libraries.

Groovy uses a curly-bracket syntax similar to Java. Groovy supports closures, multiline strings, and expressions embedded in strings.

And much of Groovy's power lies in its ASTtransformations, triggered through annotations.

Question: Why Groovy Is Gaining Popularity?

Here are few reasons for popularity of Groovy

- Familiar OOP language syntax.
- Extensive stock of various Java libraries

- Increased expressivity (type less to do more)
- Dynamic typing (lets you code more quickly, at least initially)
- Closures
- Native associative array/key-value mapping support (you can create an associative array literal)
- String interpolation (cleaner creation of strings displaying values)
- Regex's being first class citizens

Question: What Is Meant By Thin Documentation In Groovy

Groovy is documented very badly. In fact the core documentation of Groovy is limitedand there is no information regarding the complex and run-time errors that happen.

Developers are largely on there own and they normally have to figure out the explanations about internal workings by themselves.

Question: How To Run Shell Commands in Groovy?

Groovy adds the `execute` method to `String` to make executing shells fairly easy

```
println "ls".execute().text
```

Question: In How Many Platforms you can use Groovy?

These are the infrastructure components where we can use groovy:

-Application Servers

-Servlet Containers

-Databases with JDBC drivers

-All other Java-based platforms

Question: Can Groovy Integrate With Non Java Based Languages?

It is possible but in this case the features are limited. Groovy cannot be made to handle all the tasks in a manner it has to.

Question: What are Pre-Requirements For Groovy?

Installing and using Groovy is easy. Groovy does not have complex system requirements. It is OS independent.

Groovy can perform optimally in every situation.There are many Java based components in Groovy,which make it even more easier to work with Java applications.

Questions: What Is Closure In Groovy?

A closure in Groovy is an open, anonymous, block of code that can take arguments, return a value and be assigned to a variable. A closure may reference variables declared in its surrounding scope. In opposition to the formal definition of a closure, `Closure` in the Groovy language can also contain free variables which are defined outside of its surrounding scope.

A closure definition follows this syntax:

```
{ [closureParameters -> ] statements }
```

Where `[closureParameters->]` is an optional comma-delimited list of parameters, and statements are 0 or more Groovy statements. The parameters look similar to a method parameter list, and these parameters may be typed or untyped.

When a parameter list is specified, the `->` character is required and serves to separate the arguments from the closure body. The *statements* portion consists of 0, 1, or many Groovy statements.

Question: What is ExpandoMeta Class In Groovy?

Through this class programmers can add properties, constructors, methods and operations in the task. It is a powerful option available in the Groovy.

By default this class cannot be inherited and users need to call explicitly. The command for this is “ExpandoMetaClass.enableGlobally()”.

Question: What Are Limitations Of Groovy?

Groovy has some limitations. They are described below

- It can be slower than the other object-oriented programming languages.
- It might need memory more than that required by other languages.
- The start-up time of groovy requires improvement. It is not that frequent.
- For using groovy, you need to have enough knowledge of Java. Knowledge of Java is important because half of groovy is based on Java.

- It might take you some time to get used to the usual syntax and default typing.
- It consists of thin documentation.

Question: How To Write HelloWorld Program In Groovy

The following is a basic Hello World program written in Groovy:

```
class Test {

static void main(String[] args) {

println('Hello World');

}

}
```

Question: How To Declare String In Groovy?

In Groovy, the following steps are needed to declare a string.

- The string is closed with single and double qotes.
- It contains Groovy Expressions noted in ${}
- Square bracket syntax may be applied like charAt(i)

Question: Differences Between Java And Groovy?

Groovy tries to be as natural as possible for Java developers. Here are all the major differences between Java and Groovy.

-Default imports

In Groovy all these packages and classes are imported by default, i.e. Developers do not have to use an explicit `import` statement to use them:

- java.io.*
- java.lang.*
- java.math.BigDecimal
- java.math.BigInteger
- java.net.*
- java.util.*
- groovy.lang.*
- groovy.util.*

-Multi-methods

In Groovy, the methods which will be invoked are chosen at runtime. This is called runtime dispatch or multi-methods. It means that the method will be chosen based on the types of the arguments at runtime. In Java, this is the opposite: methods are chosen at compile time, based on the declared types.

-Array initializers

In Groovy, the `{ … }` block is reserved for closures. That means that you cannot create array literals with this syntax:

```
int[] arraySyntex = { 6, 3, 1}
```

You actually have to use:

```
int[] arraySyntex = [1,2,3]
```

-ARM blocks

ARM (Automatic Resource Management) block from Java 7 are not supported in Groovy. Instead, Groovy provides various methods relying on closures, which have the same effect while being more idiomatic.

-GStrings

As double-quoted string literals are interpreted as `GString` values, Groovy may fail with compile error or produce subtly different code if a class with `String` literal containing a dollar character is compiled with Groovy and Java compiler.

While typically, Groovy will auto-cast between `GString` and `String` if an API declares the type of a parameter, beware of Java APIs that accept an `Object` parameter and then check the actual type.

-String and Character literals

Singly-quoted literals in Groovy are used for `String` , and double-quoted result in `String` or `GString` , depending whether there is interpolation in the literal.

```
assert 'c'.getClass()==String

assert "c".getClass()==String

assert "c${1}".getClass() in GString
```

Groovy will automatically cast a single-character `String` to `char` only when assigning to a variable of type `char` . When calling methods with arguments of type `char` we need to either cast explicitly or make sure the value has been cast in advance.

```
char a='a'

assert Character.digit(a, 16)==10 : 'But Groovy does boxing' assert
Character.digit((char) 'a', 16)==10


try {

    assert Character.digit('a', 16)==10

    assert false: 'Need explicit cast'
```

```
} catch(MissingMethodException e) {

}
```

Groovy supports two styles of casting and in the case of casting to `char` there are subtle differences when casting a multi-char strings. The Groovy style cast is more lenient and will take the first character, while the C-style cast will fail with exception.

```
        //  for single char strings, both are the same
        assert ((char) "c").class==Character

assert ("c" as char).class==Character

        //  for multi char strings they are not

try {

    ((char) 'cx') == 'c'

    assert false: 'will fail - not castable'

} catch(GroovyCastException e) {

}

assert ('cx' as char) == 'c' assert
'cx'.asType(char) == 'c'
```

-Behaviour of `==`

In Java `==` means equality of primitive types or identity for objects. In Groovy `==` translates to `a.compareTo(b)==0` , if they are `Comparable` ,

and `a.equals(b)` otherwise. To check for identity, there is `is` . E.g. `a.is(b)` .

Question: How To Test Groovy Application?

The Groovy programming language comes with great support for writing tests. In addition to the language features and test integration with state-of-the-art testing libraries and frameworks.

The Groovy ecosystem has born a rich set of testing libraries and frameworks.

Groovy Provides following testing capabilities

Junit Integrations

Spock for specifications

Geb for Functional Test

Groovy also has excellent built-in support for a range of mocking and stubbing alternatives.

When using Java, dynamic mocking frameworks are very popular.

A key reason for this is that it is hard work creating custom hand-crafted mocks using Java.

Such frameworks can be used easily with Groovy.

Question: What Are Power Assertions In Groovy?

Writing tests means formulating assumptions by using assertions. In Java this can be done by using the `assert` keyword. But Groovy comes with a *powerful variant* of `assert` also known as *power assertion statement*.

Groovy's power `assert` differs from the Java version in its output given the boolean expression validates to `false` :

```
def x = 1

assert x == 2

    //  Output:
    //
    //  Assertion failed:
    //  assert x == 2
    //  | |
    //  1 false
```

This section shows the std-err output

The `java.lang.AssertionError` that is thrown whenever the assertion can not be validated successfully, contains an extended version of the original exception message. The power assertion output shows evaluation results from the outer to the inner expression. The power assertion statements true power unleashes in complex Boolean statements, or statements with collections or other `toString` -enabled classes:

```
def x = [1,2,3,4,5]

assert (x << 6) == [6,7,8,9,10]

        //  Output:
        //
        //  Assertion failed:
        //  assert (x << 6) == [6,7,8,9,10]
//                |  |          |
//                |  |          false
                  // | [1, 2, 3, 4, 5, 6]
                  // [1, 2, 3, 4, 5, 6]
```

Question: Can We Use Design Patterns In Groovy?

Design patterns can also be used with Groovy. Here are important points

- Some patterns carry over directly (and can make use of normal Groovy syntax improvements for greater readability)
- Some patterns are no longer required because they are built right into the language or because Groovy supports a better way of achieving the intent of the pattern
- some patterns that have to be expressed at the design level in other languages can be implemented directly in Groovy (due to the way Groovy can blur the distinction between design and implementation)

Question: How To Parse And Produce JSON Object In Groovy?

Groovy comes with integrated support for converting between Groovy objects and JSON. The classes dedicated to JSON serialisation and parsing are found in the `groovy.json` package.

`JsonSlurper` is a class that parses JSON text or reader content into Groovy data structures (objects) such as maps, lists and primitive types like `Integer` , `Double` , `Boolean` and `String` .

The class comes with a bunch of overloaded `parse` methods plus some special methods such as `parseText` , `parseFile` and others

---

Question: What Is Difference Between XmlParser And XmlSluper?

XmlParser and XmlSluper are used for parsing XML with Groovy. Both have the same approach to parse an xml.

Both come with a bunch of overloaded parse methods plus some special methods such as `parseText` , parseFile and others.

XmlSlurper

```
def text = '''
    <list>
        <technology>
            <name>Groovy</name>
        </technology>
    </list>
'''

def list = new XmlSlurper().parseText(text)

assert list instanceof groovy.util.slurpersupport.GPathResult assert
list.technology.name == 'Groovy'
```

Parsing the XML an returning the root node as a GPathResult

Checking we’re using a GPathResult

Traversing the tree in a GPath style

XmlParser

```
def text = '''
    <list>
        <technology>
            <name>Groovy</name>
        </technology>
    </list>
'''

def list = new XmlParser().parseText(text)

assert list instanceof groovy.util.Node assert
list.technology.name.text() == 'Groovy'
```

Parsing the XML an returning the root node as a Node

Checking we're using a Node

Traversing the tree in a GPath style

Let's see the similarities between `XMLParser` and `XMLSlurper` first:

- Both are based on `SAX` so they both are low memory footprint
- Both can update/transform the XML

But they have key differences:

- `XmlSlurper` evaluates the structure lazily. So if you update the xml you'll have to evaluate the whole tree again.

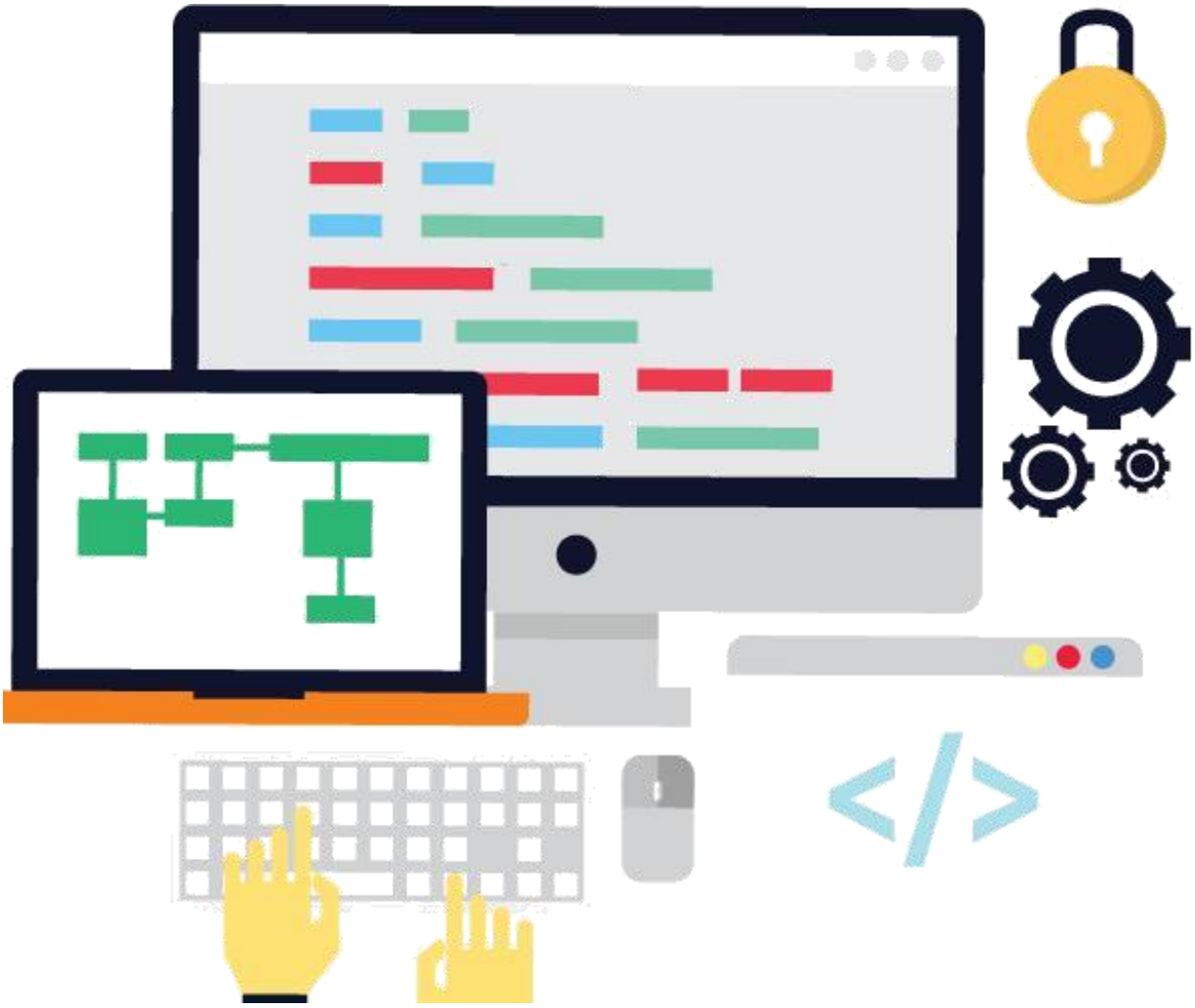
- `XmlSlurper` returns `GPathResult` instances when parsing XML
- `XmlParser` returns `Node` objects when parsing XML

When to use one or the another?

- If you want to transform an existing document to another then `XmlSlurper` will be the choice

- If you want to update and read at the same time then `XmlParser` is the choice.

Maven DevOps Interview Questions



Question: What is Maven?

Maven is a build automation tool used primarily for Java projects. Maven addresses two aspects of building software:

First: It describes how software is built

Second: It describes its dependencies.

Unlike earlier tools like Apache Ant, it uses conventions for the build procedure, and only exceptions need to be written down.

An XML file describes the software project being built, its dependencies on other external modules and components, the build order, directories, and required plug-ins.

It comes with pre-defined targets for performing certain well-defined tasks such as compilation of code and its packaging.

Maven dynamically downloads Java libraries and Maven plug-ins from one or more repositories such as the Maven 2 Central Repository, and stores them in a local cache.

This local cache of downloaded artifacts can also be updated with artifacts created by local projects. Public repositories can also be updated.

Question: What Are Benefits Of Maven?

- One of the biggest benefit of Maven is that its design regards all projects as having a certain structure and a set of supported task work-flows.
- Maven has quick project setup, no complicated build.xml files, just a POM and go
- All developers in a project use the same jar dependencies due to centralized
- POM. In Maven getting a number of reports and metrics for a project "for free"
- It reduces the size of source distributions, because jars can be pulled from a central location
- Maven lets developers get your package dependencies easily
- With Maven there is no need to add jar files manually to the class path

Question: What Are Build Life cycles In Maven?

Build lifecycle is a list of named phases that can be used to give order to goal execution. One of Maven's standard life cycles is the *default lifecycle*, which includes the following phases, in this order

```
1  validate
2  generate-sources
3  process-sources
4  generate-resources
5  process-resources
6  compile
7  process-test-sources
8  process-test-resources
9  test-compile
10  test
11  package
```

```
12 install
13 deploy
```

Question: What Is Meant By Build Tool?

Build tools are programs that automate the creation of executable applications from source code. Building incorporates compiling, linking and packaging the code into a usable or executable form.

In small projects, developers will often manually invoke the build process. This is not practical for larger projects.

Where it is very hard to keep track of what needs to be built, in what sequence and what dependencies there are in the building process. Using an automation tool like Maven, Gradle or ANT allows the build process to be more consistent.

Question: What Is Dependency Management Mechanism In Gradle?

Maven's dependency-handling mechanism is organized around a coordinate system identifying individual artifacts such as software libraries or modules.

For example if a project needs Hibernate library. It has to simply declare Hibernate's project coordinates in its POM.

Maven will automatically download the dependency and the dependencies that Hibernate itself needs and store them in the user's local repository.

Maven 2 Central Repository is used by default to search for libraries, but developers can configure the custom repositories to be used (e.g., company-private repositories) within the POM.

Question: What Is Central Repository Search Engine?

The Central Repository Search Engine, can be used to find out coordinates for different open-source libraries and frameworks.

Question: What are Plugins In Maven?

Most of Maven's functionality is in plugins. A plugin provides a set of goals that can be executed using the following syntax:

```
mvn [plugin-name]:[goal-name]
```

For example, a Java project can be compiled with the compiler-plugin's compile-goal by running `mvn compiler:compile` . There are Maven plugins for building, testing, source control management, running a web server, generating Eclipse project files, and much more.

Plugins are introduced and configured in a <plugins>-section of a `pom.xml` file. Some basic plugins are included in every project by default, and they have sensible default settings.

Questions: What Is Difference Between Maven And ANT?

Ant	Maven
Ant is a tool box.	Maven is a framework.
There is no life cycle.	There is life cycle.

Ant doesn't have formal conventions.	Maven has a convention to place source code, compiled code etc.
Ant is procedural.	Maven is declarative.
The ant scripts are not reusable.	The maven plugins are reusable.

Question: What is POM In Maven?

A Project Object Model (POM) provides all the configuration for a single project. General configuration covers the project's name, its owner and its dependencies on other projects.

One can also configure individual phases of the build process, which are implemented as plugins.

For example, one can configure the compiler-plugin to use Java version 1.5 for compilation, or specify packaging the project even if some unit tests fail.

Larger projects should be divided into several modules, or sub-projects, each with its own POM. One can then write a root POM through which one can compile all the modules with a single command. POMs can also inherit configuration from other POMs. All POMs inherit from the Super POM by default. The Super POM provides default configuration, such as default source directories, default plugins, and so on.

Question: What Is Maven Archetype?

Archetype is a Maven project templating toolkit. An archetype is defined as an original pattern or model from which all other things of the same kind are made.

Question: What Is Maven Artifact?

In Maven artifact is simply a file or JAR that is deployed to a Maven repository. An artifact has

-Group ID

-Artifact ID

-Version string. The three together uniquely identify the artifact. All the project dependencies are specified as artifacts.

Question: What Is Goal In Maven?

In Maven a goal represents a specific task which contributes to the building and managing

of a project.

It may be bound to 1 or many build phases. A goal not bound to any build phase could be executed outside of the build lifecycle by its direct invocation.

Question: What Is Build Profile?

In Maven a build profile is a set of configurations. This set is used to define or override default behaviour of Maven build.

Build profile helps the developers to customize the build process for different environments. For example you can set profiles for Test, UAT, Pre-prod and Prod environments each with its own configurations etc.

Question: What Are Build Phases In Maven?

There are 6 build phases. -Validate -Compile -Test -Package -Install -Deploy

Question: What Is Target, Source & Test Folders In Mavn?

Target: folder holds the compiled unit of code as part of the build process.
Source: folder usually holds java source codes. **Test:** directory contains all the unit testing codes.

Question: What Is Difference Between Compile & Install?

Compile: is used to compile the source code of the project **Install:** installs the package into the local repository, for use as a dependency in other projects locally.Design patterns can also be used with Groovy. Here are important points

Question: How To Activate Maven Build Profile?

A Maven Build Profile can be activated in following ways

- Using command line console input.
- By using Maven settings.
- Based on environment variables (User/System variables).

Linux DevOps Interview Questions

Question: What is Linux?

Linux is the best-known and most-used open source operating system. As an operating system, Linux is a software that sits underneath all of the other software on a computer,

receiving requests from those programs and relaying these requests to the computer’s hardware.

In many ways, Linux is similar to other operating systems such as Windows, OS X, or iOS

But Linux also is different from other operating systems in many important ways.

First, and perhaps most importantly, Linux is open source software. The code used to create Linux is free and available to the public to view, edit, and—for users with the appropriate skills—to contribute to.

Linux operating system is consist of 3 components which are as below:

- **Kernel:** Linux is a monolithic kernel that is free and open source software that is responsible for managing hardware resources for the users.
- **System Library:** System Library plays a vital role because application programs access Kernels feature using system library.
- **System Utility:** System Utility performs specific and individual level tasks.

Question: What Is Difference Between Linux & Unix?

Unix and Linux are similar in many ways, and in fact, Linux was originally created to be similar to Unix.

Both have similar tools for interfacing with the systems, programming tools, filesystem layouts, and other key components.

However, Unix is not free. Over the years, a number of different operating systems have been created that attempted to be “unix-like” or “unix-compatible,” but Linux has been the most successful, far surpassing its predecessors in popularity.

Question: What Is BASH?

BASH stands for **Bourne Again Shell**. BASH is the UNIX shell for the GNU operating system. So, BASH is the command language interpreter that helps you to enter your input, and thus you can retrieve information.

In a straightforward language, BASH is a program that will understand the data entered by the user and execute the command and gives output.

Question: What Is CronTab?

The crontab (short for "cron table") is a list of commands that are scheduled to run at regular time intervals on computer system. The **crontab** command opens the crontab for editing, and lets you add, remove, or modify scheduled tasks.

The daemon which reads the crontab and executes the commands at the right time is called cron. It's named after Kronos, the Greek god of time.

Command syntax

crontab [-u *user*] *file*

crontab [-u *user*] [-l | -r | -e] [-i] [-s]

Question: What Is Daemon In Linux?

A **daemon** is a type of program on Linux operating systems that runs unobtrusively in the background, rather than under the direct control of a user, waiting to be activated by the occurrence of a specific event or condition

Unix-like systems typically run numerous daemons, mainly to accommodate requests for services from other computers on a network, but also to respond to other programs and to hardware activity.

Examples of actions or conditions that can trigger daemons into activity are a specific time or date, passage of a specified time interval, a file landing in a particular directory, receipt of an e-mail or a Web request made through a particular communication line.

It is not necessary that the perpetrator of the action or condition be aware that a daemon is *listening*, although programs frequently will perform an action only because they are aware that they will implicitly arouse a daemon.

Question: What Is Process In Linux?

Daemons are usually instantiated as *processes*. A process is an *executing* (i.e., running) instance of a program.

Processes are managed by the *kernel* (i.e., the core of the operating system), which assigns each a unique *process identification number* (PID).

There are three basic types of processes in Linux:

-Interactive: Interactive processes are run interactively by a user at the *command line*

-Batch: Batch processes are submitted from a queue of processes and are not associated with the command line; they are well suited for performing recurring tasks when system usage is otherwise low.

-Daemon: Daemons are recognized by the system as any processes whose *parent process* has a PID of one

---

Question: What Is CLI In Linux?

CLI (Command Line Interface) is a type of *human-computer interface* that relies solely on textual input and output.

That is, the entire display screen, or the currently active portion of it, shows

only characters (and no images), and input is usually performed entirely with a keyboard.

Questions: What Is Linux Kernel?

A kernel is the lowest level of easily replaceable software that interfaces with the hardware in your computer.

It is responsible for interfacing all of your applications that are running in “user mode” down

to the physical hardware, and allowing processes, known as servers, to get information from each other using inter-process communication (IPC).

There are three types of Kernals

Microkernel:A microkernel takes the approach of only managing what it has to: CPU, memory, and IPC. Pretty much everything else in a computer can be seen as an accessory and can be handled in user mode.

Monolithic Kernel: Monolithic kernels are the opposite of microkernels because they encompass not only the CPU, memory, and IPC, but they also include things like device drivers, file system management, and system server calls

Hybrid Kernel: Hybrid kernels have the ability to pick and choose what they want to run in user mode and what they want to run in supervisor mode. Because the Linux kernel is monolithic, it has the largest footprint and the most complexity over the other types of kernels. This was a design feature which was under quite a bit of debate in the early days of Linux and still carries some of the same design flaws that monolithic kernels are inherent to have.

Question: What Is Partial Backup In Linux?

Partial backup refers to selecting only a portion of file hierarchy or a single partition to back up.

Question: What Is Root Account?

The root account a system administrator account. It provides you full access and control of the system.

Admin can create and maintain user accounts, assign different permission for each account etc

---

Question: What Is Difference Between Cron and **Anacron**?

One of the main difference between cron and anacron jobs is that cron works on the system that are running continuously.

While anacron is used for the systems that are not running continuously.

1. Other difference between the two is cron jobs can run every minute, but anacron jobs can be run only once a day.
2. Any normal user can do the scheduling of cron jobs, but the scheduling of anacron jobs can be done by the superuser only.

3. Cron should be used when you need to execute the job at a specific time as per the given time in cron, but anacron should be used in when there is no any restriction for the timing and can be executed at any time.
4. If we think about which one is ideal for servers or desktops, then cron should be used for servers while anacron should be used for desktops or laptops.

Question: What Is Linux Loader?

Linux Loader is a boot loader for Linux operating system. It loads Linux into into the main memory so that it can begin its operations.

Question: What Is Swap Space?

Swap space is the amount of physical memory that is allocated for use by Linux to hold some concurrent running programs temporarily.

This condition usually occurs when Ram does not have enough memory to support all concurrent running programs.

This memory management involves the swapping of memory to and from physical storage.

Question: What Are Linux Distributors?

There are around six hundred Linux distributors. Let us see some of the important ones

- ***UBuntu:*** It is a well known Linux Distribution with a lot of pre-installed apps and easy to use repositories libraries. It is very easy to use and works like MAC operating system.

- ***Linux Mint:*** It uses cinnamon and mate desktop. It works on windows and should be used by newcomers.

- ***Debian:*** It is the most stable, quicker and user-friendly Linux Distributors.
- ***Fedora:*** It is less stable but provides the latest version of the software. It has GNOME3 desktop environment by default.
- ***Red Hat Enterprise:*** It is to be used commercially and to be well tested before release. It usually provides the stable platform for a long time.
- ***Arch Linux:*** Every package is to be installed by you and is not suitable for the beginners.

Question: Why Do Developers Use MD5?

MD5 is an encryption method so it is used to encrypt the passwords before saving.

Question: What Are File Permissions In Linux?

There are 3 types of permissions in Linux

- **Read:** User can read the file and list the directory.
- **Write:** User can write new files in the directory .
- **Execute:** User can access and run the file in a directory.

Question: Memory Management In Linux?

It is always required to keep a check on the memory usage in order to find out whether the user is able to access the server or the resources are adequate. There are roughly 5 methods that determine the total memory used by the Linux.

This is explained as below

- ***Free command*:** This is the most simple and easy to use the command to check memory usage. For example: '$ free –m', the option 'm' displays all the data in MBs.
- **/proc/meminfo:** The next way to determine the memory usage is to read /proc/meminfo file. For example: '$ cat /proc/meminfo'
- ***Vmstat*:** This command basically lays out the memory usage statistics. For example: '$ vmstat –s'
- ***Top command*:** This command determines the total memory usage as well as also monitors the RAM usage.
- ***Htop*:** This command also displays the memory usage along with other details.

Question: Granting Permissions In Linux?

System administrator or the owner of the file can grant permissions using the ‘chmod’ command. Following symbols are used while writing permissions

chmod +x

Question: What Are Directory Commands In Linux?

Here are few important directory commands in Linux

- ***pwd:*** It is a built-in command which stands for **‘print working directory’**. It displays the current working location, working path starting with / and directory of the user. Basically, it displays the full path to the directory you are currently in. ***ls:***
- This command list out all the files in the directed folder.

- ***cd:*** This stands for ‘change directory’. This command is used to change to the

directory you want to work from the present directory. We just need to type cd followed by the directory name to access that particular directory. ***mkdir:*** This
- command is used to create an entirely new directory.

- ***rmdir:*** This command is used to remove a directory from the system.

Question: What Is Shell Script In Linux?

In the simplest terms, a shell script is a file containing a series of commands.

The shell reads this file and carries out the commands as though they have been entered directly on the command line.

The shell is somewhat unique, in that it is both a powerful command line interface to the system and a scripting language interpreter.

As we will see, most of the things that can be done on the command line can be done in scripts, and most of the things that can be done in scripts can be done on the command line.

We have covered many shell features, but we have focused on those features most often used directly on the command line.

The shell also provides a set of features usually (but not always) used when writing programs.

Question: Which Tools Are Used For Reporting Statistics In Linux?

Some of the popular and frequently used system resource generating tools available on the Linux platform are

- vmstat

- netstat
- iostat
- **ifstat**
- mpstat.

These are used for reporting statistics from different system components such as virtual memory, network connections and interfaces, CPU, input/output devices and more.

Question: What Is Dstat In Linux?

---

dstat is a powerful, flexible and versatile tool for generating Linux system resource statistics, that is a replacement for all the tools mentioned in above question.

It comes with extra features, counters and it is highly extensible, users with Python knowledge can build their own plugins.

Features of dstat:

1. Joins information from vmstat, netstat, iostat, ifstat and mpstat tools
2. Displays statistics simultaneously
3. Orders counters and highly-extensible
4. Supports summarizing of grouped block/network devices
5. Displays interrupts per device
6. Works on accurate timeframes, no timeshifts when a system is stressed
7. Supports colored output, it indicates different units in different colors
8. Shows exact units and limits conversion mistakes as much as possible
9. Supports exporting of CSV output to Gnumeric and Excel documents

Question: Types Of Processes In Linux?

There are fundamentally two types of processes in Linux:

- **Foreground processes** (also referred to as interactive processes) – these are initialized and controlled through a terminal session. In other words, there has to be a user connected to the system to start such processes; they haven't started automatically as part of the system functions/services.
- **Background processes** (also referred to as non-interactive/automatic processes) – are processes not connected to a terminal; they don't expect any user input.

Question: Creatin Of Processes In Linux?

A new process is normally created when an existing process makes an exact copy of itself in memory.

The child process will have the same environment as its parent, but only the process ID number is different.

There are two conventional ways used for creating a new process in Linux:

- **Using The System() Function** – this method is relatively simple, however, it's inefficient and has significantly certain security risks.
- **Using fork() and exec() Function** – this technique is a little advanced but offers greater flexibility, speed, together with security.

Question: Creation Of Processes In Linux?

Because Linux is a multi-user system, meaning different users can be running various programs on the system, each running instance of a program must be identified uniquely by the kernel.

And a program is identified by its process ID (**PID**) as well as it's parent processes ID (**PPID**), therefore processes can further be categorized into:

- **Parent processes** – these are processes that create other processes during run-time.
- **Child processes** – these processes are created by other processes during run-time.

Question: What Is Init Process Linux?

I**nit** process is the mother (parent) of all processes on the system, it's the first program that is executed when the Linux system boots up; it manages all other processes on the system. It is started by the kernel itself, so in principle it does not have a parent process.

The init process always has process ID of **1**. It functions as an adoptive parent for all orphaned processes.

You can use the **pidof command** to find the ID of a process:

```
# pidof systemd
# pidof top
# pidof httpd
```

Find Linux Process ID

To find the process ID and parent process ID of the current shell, run:

```
$ echo $$
$ echo $PPID
```

Question: What Are Different States Of A Processes In Linux?

During execution, a process changes from one state to another depending on its environment/circumstances. In Linux, a process has the following possible states:

- **Running** – here it's either running (it is the current process in the system) or it's ready to run (it's waiting to be assigned to one of the CPUs).
- **Waiting** – in this state, a process is waiting for an event to occur or for a system resource. Additionally, the kernel also differentiates between two types of waiting processes; interruptible waiting processes – can be interrupted by signals and uninterruptible waiting processes – are waiting directly on hardware conditions and cannot be interrupted by any event/signal.
- **Stopped** – in this state, a process has been stopped, usually by receiving a signal.

  For instance, a process that is being debugged.

- **Zombie** – here, a process is dead, it has been halted but it's still has an entry in the process table.

Question: How To View Active Processes In Linux?

There are several Linux tools for viewing/listing running processes on the system, the two traditional and well known are ps and top commands:

1. ps Command

It displays information about a selection of the active processes on the system as shown below:

#ps

#ps -e] head

2. top – System Monitoring Tool

top is a powerful tool that offers you a dynamic real-time view of a running system as shown in the screenshot below:

#top

3. glances – System Monitoring Tool

glances is a relatively new system monitoring tool with advanced features:

#glances

Question: How To Control Process?

Linux also has some commands for controlling processes such as kill, pkill, pgrep and killall, below are a few basic examples of how to use them:

```
$ pgrep -u tecmint top
$ kill 2308
$ pgrep -u tecmint top
$ pgrep -u tecmint glances
$ pkill glances
$ pgrep -u tecmint glances
```

Question: Can We Send signals To Processes In Linux?

The fundamental way of controlling processes in Linux is by sending signals to them. There are multiple signals that you can send to a process, to view all the signals run:

```
$ kill -l
```

List All Linux Signals

To send a signal to a process, use the kill, pkill or pgrep commands we mentioned earlier on. But programs can only respond to signals if they are programmed to recognize those signals.

And most signals are for internal use by the system, or for programmers when they write code. The following are signals which are useful to a system user:

- **SIGHUP 1** – sent to a process when its controlling terminal is closed.
- **SIGINT 2** – sent to a process by its controlling terminal when a user interrupts the process by pressing `[Ctrl+C]` .
- **SIGQUIT 3** – sent to a process if the user sends a quit signal `[Ctrl+D]` .
- **SIGKILL 9** – this signal immediately terminates (kills) a process and the process will not perform any clean-up operations.
- **SIGTERM 15** – this a program termination signal (kill will send this by default).
- **SIGTSTP 20** – sent to a process by its controlling terminal to request it to stop (terminal stop); initiated by the user pressing `[Ctrl+Z]` .

Question: How To Change Priority Of A Processes In Linux?

On the Linux system, all active processes have a priority and certain nice value. Processes with higher priority will normally get more CPU time than lower priority processes.

However, a system user with root privileges can influence this with the **nice** and **renice** commands.

From the output of the top command, the NI shows the process nice value:

```
$ top
```

List Linux Running Processes

Use the **nice** command to set a nice value for a process. Keep in mind that normal users can attribute a nice value from zero to 20 to processes they own. Only the root user can use negative nice values.

To **renice** the priority of a process, use the **renice** command as follows:

```
$  renice +8  2687
$  renice +8  2103
```

GIT DevOps Interview Questions

Question: What is Git?

Git is a version control system for tracking changes in computer files and coordinating work on those files among multiple people.

It is primarily used for source code management in software development but it can be used to keep track of changes in any set of files.

As a distributed revision control system it is aimed at speed, data integrity, and support for distributed, non-linear workflows.

By far, the most widely used modern version control system in the world today is Git. Git is a mature, actively maintained open source project originally developed in 2005 by Linus Torvald. Git is an example of a Distributed Version Control System, In Git, every developer's working copy of the code is also a repository that can contain the full history of all changes.

Question: What Are Benefits Of GIT?

Here are some of the advantages of using Git

- Ease of use
- Data redundancy and replication
- High availability
- Superior disk utilization and network performance
- Only one .git directory per repository
- Collaboration friendly
- Any kind of projects from large to small scale can use GIT

Question: What Is Repository In GIT?

The purpose of Git is to manage a project, or a set of files, as they change over time. Git stores this information in a data structure called a repository. A git **repository** contains, among other things, the following:

- A set of **commit objects**.
- A set of references to commit objects, called **heads**.

The Git repository is stored in the same directory as the project itself, in a subdirectory called `.git`. Note differences from central-repository systems like CVS or Subversion:

- There is only one `.git` directory, in the root directory of the project.
- The repository is stored in files alongside the project. There is no central server repository.

Question: What Is Staging Area In GIT?

Staging is a step before the commit process in git. That is, a commit in git is performed in two steps:

-Staging and

-Actual commit

As long as a change set is in the staging area, git allows you to edit it as you like

(replace staged files with other versions of staged files, remove changes from staging, etc.)

Question: What Is GIT STASH?

Often, when you've been working on part of your project, things are in a messy state and you want to switch branches for a bit to work on something else.

The problem is, you don't want to do a commit of half-done work just so you can get back to this point later. The answer to this issue is the `git stash` command. Stashing takes the dirty state of your working directory — that is, your modified tracked files and staged changes — and saves it on a stack of unfinished changes that you can reapply at any time.

Question: How To Revert Commit In GIT?

Given one or more existing commits, revert the changes that the related patches introduce, and record some new commits that record them. This requires your working tree to be clean (no modifications from the HEAD commit).

git-revert - Revert some existing commits

SYNOPSIS

```
git revert [--[no-]edit] [-n] [-m parent-number] [-s] [-S[<keyid>]] <commit>…
```

```
git revert --continue

git revert --quit

git revert --abort
```

Question: How To Delete Remote Repository In GIT?

Use the `git remote rm` command to remove a remote URL from your repository.

The `git remote rm` command takes one argument:

> A remote name, for example, `destination`

Questions: What Is GIT Stash Drop?

In case we do not need a specific stash, we use git stash drop command to remove it from the list of stashes.

By default, this command removes to latest added stash

To remove a specific stash we specify as argument in the git stash drop <stashname> command.

Question: What Is Difference Between GIT and Subversion?

Here is a summary of Differences between GIT and Subversion

- Git is a distributed VCS; SVN is a non-distributed VCS.
- Git has a centralized server and repository; SVN does not have a centralized server or repository.
- The content in Git is stored as metadata; SVN stores files of content.
- Git branches are easier to work with than SVN branches.
- Git does not have the global revision number feature like SVN has.
- Git has better content protection than SVN.
- Git was developed for Linux kernel by Linus Torvalds; SVN was developed by CollabNet, Inc.
- Git is distributed under GNU, and its maintenance overseen by Junio

  Hamano; Apache Subversion, or SVN, is distributed under the open source license.

Question: What Is Difference Between GIT Fetch & GIT Pull?

GIT fetch – It downloads only the new data from the remote repository and does not integrate any of the downloaded data into your working files. Providing a view of the data is all it does.

GIT pull – It downloads as well as merges the data from the remote repository into the local working files.

This may also lead to merging conflicts if the user's local changes are not yet committed.

Using the "GIT stash" command hides the local changes.

Question: **What is Git fork? How to create tag?**

A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

A fork is really a Github (not Git) construct to store a clone of the repo in your user account.

As a clone, it will contain all the branches in the main repo at the time you made the fork.

Create Tag:

- Click the releases link on our repository page.
- Click on Create a new release or Draft a new release.
- Fill out the form fields, then click Publish release at the bottom.
- After you create your tag on GitHub, you might want to fetch it into your local repository too: git fetch.

Question: What is difference between fork and branch?

A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

A fork is really a Github (not Git) construct to store a clone of the repo in your user account.

As a clone, it will contain all the branches in the main repo at the time you made the fork.

Question: What Is Cherry Picking In GIT?

Cherry picking in git means to choose a commit from one branch and apply it onto another.

This is in contrast with other ways such as merge and rebase which normally applies many commits onto a another branch.

Make sure you are on the branch you want apply the commit to. git checkout master

Execute the following:

```
git cherry-pick <commit-hash>
```

Question: What Language GIT Is Written In?

Much of Git is written in C, along with some BASH scripts for UI wrappers and other bits.

Question: How To Rebase Master In GIT?

Rebasing is the process of moving a branch to a new base commit.The golden rule of git rebase is to never use it on public branches.

The only way to synchronize the two master branches is to merge them back together, resulting in an extra merge commit and two sets of commits that contain the same changes.

Question: **What is ‘head’ in git and how many heads can be created in a repository?**

There can be any number of heads in a GIT repository. By default there is one head known as HEAD in each repository in GIT.

HEAD is a ref (reference) to the currently checked out commit. In normal states, it's actually a symbolic ref to the branch user has checked out.

if you look at the contents of .**git**/**HEAD** you'll see something like "ref: refs/**heads**/master". The branch itself is a reference to the commit at the tip of the branch

---

Question: Name some GIT commands and also explain their functions?

Here are some most important GIT commands

- **GIT diff** – It shows the changes between commits, commits and working tree.
- **GIT status** – It shows the difference between working directories and index.
- **GIT stash applies** – It is used to bring back the saved changes on the working directory.
- **GIT rm** – It removes the files from the staging area and also of the disk.
- **GIT log** – It is used to find the specific commit in the history.
- **GIT add** – It adds file changes in the existing directory to the index.
- **GIT reset** – It is used to reset the index and as well as the working directory to the state of the last commit.
- **GIT checkout** – It is used to update the directories of the working tree with those from another branch without merging.
- **GIT Is tree** – It represents a tree object including the mode and the name of each item.
- **GIT** instaweb – It automatically directs a web browser and runs the web server with an interface into your local repository.

Question: What is a “conflict” in GIT and how is it resolved?

When a commit that has to be merged has some changes in one place, which also has the changes of current commit, then the conflict arises.

The GIT will not be able to predict which change will take the precedence. In order to resolve the conflict in GIT: we have to edit the files to fix the conflicting changes and then add the resolved files by running the “GIT add” command; later on, to commit the

repaired merge run the “GIT commit” command. GIT identifies the position and sets the parents of the commit correctly.

Question: How To Migrate From Subversion To GIT?

SubGIT is a tool for smooth and stress-free subversion to GIT migration and also a solution for a company-wide subversion to GIT migration that is:

- It allows to make use of all GIT and subversion features.
- It provides genuine stress-free migration experience.
- It doesn’t require any change in the infrastructure that is already placed.
- It is considered to be much better than GIT-SVN

Question: What Is Index In GIT?

The index is a single, large, binary file in under .git folder, which lists all files in the current branch, their sha1 checksums, time stamps and the file name. Before completing the commits, it is formatted and reviewed in an intermediate area known as Index also known as the staging area.

Question: What is a bare Git repository?

A bare Git repository is a repository that is created without a Working Tree.

```
git init --bare
```

Question: WHow do you revert a commit that has already been pushed and made public??

One or more commits can be reverted through the use of *git revert*. This command, in essence, creates a new commit with patches that cancel out the changes introduced in specific commits.

In case the commit that needs to be reverted has already been published or changing the repository history is not an option, *git revert* can be used to revert commits. Running the following command will revert the last two commits:

```
git revert HEAD~2..HEAD
```

Alternatively, one can always checkout the state of a particular commit from the past, and commit it anew.

Question: How do you squash last N commits into a single commit?

Squashing multiple commits into a single commit will overwrite history, and should be done with caution. However, this is useful when working in feature branches.

To squash the last N commits of the current branch, run the following command (with {N} replaced with the number of commits that you want to squash):

```
git rebase -i HEAD~{N}
```

Upon running this command, an editor will open with a list of these N commit messages, one per line.

Each of these lines will begin with the word “pick”. Replacing “pick” with “squash” or “s” will tell Git to combine the commit with the commit before it.

To combine all N commits into one, set every commit in the list to be squash except the first one.

Upon exiting the editor, and if no conflict arises, *git rebase* will allow you to create a new commit message for the new combined commit.

Question: What is a conflict in git and how can it be resolved?

A conflict arises when more than one commit that has to be merged has some change in the same place or same line of code.

Git will not be able to predict which change should take precedence. This is a git conflict.

To resolve the conflict in git, edit the files to fix the conflicting changes and then add the resolved files by running `git add .`

After that, to commit the repaired merge, run `git commit`. Git remembers that you are in the middle of a merge, so it sets the parents of the commit correctly.

Question: How To Setup A Script To Run Every Time a Repository Receives New Commits Through Push?

To configure a script to run every time a repository receives new commits through push, one needs to define either a pre-receive, update, or a post-receive hook depending on when exactly the script needs to be triggered.

Pre-receive hook in the destination repository is invoked when commits are pushed to it.

Any script bound to this hook will be executed before any references are updated.

This is a useful hook to run scripts that help enforce development policies.

Update hook works in a similar manner to pre-receive hook, and is also triggered before any updates are actually made.

However, the update hook is called once for every commit that has been pushed to the destination repository.

Finally, post-receive hook in the repository is invoked after the updates have been accepted into the destination repository.

This is an ideal place to configure simple deployment scripts, invoke some continuous integration systems, dispatch notification emails to repository maintainers, etc.

Hooks are local to every Git repository and are not versioned. Scripts can either be created within the hooks directory inside the ".git" directory, or they can be created elsewhere and links to those scripts can be placed within the directory.

Question: What Is Commit Hash?

In Git each commit is given a unique hash. These hashes can be used to identify the corresponding commits in various scenarios (such as while trying to checkout a particular state of the code using the *git checkout {hash}* command).

Additionally, Git also maintains a number of aliases to certain commits, known as refs.

Also, every tag that you create in the repository effectively becomes a ref (and that is exactly why you can use tags instead of commit hashes in various git commands).

Git also maintains a number of special aliases that change based on the state of the repository, such as HEAD, FETCH_HEAD, MERGE_HEAD, etc.

Git also allows commits to be referred as relative to one another. For example, HEAD~1 refers to the commit parent to HEAD, HEAD~2 refers to the grandparent of HEAD, and so on.

In case of merge commits, where the commit has two parents, ^ can be used to select one of the two parents, e.g. HEAD^2 can be used to follow the second parent.

And finally, refspecs. These are used to map local and remote branches together.

However, these can be used to refer to commits that reside on remote branches allowing one to control and manipulate them from a local Git environment.

Question: What Is Conflict In GIT?

A conflict arises when more than one commit that has to be merged has some change in the same place or same line of code.

Git will not be able to predict which change should take precedence. This is a git conflict.To resolve the conflict in git, edit the files to fix the conflicting changes and then add the resolved files by running `git add .` After that, to commit the repaired merge, run `git commit` . Git remembers that you are in the middle of a merge, so it sets the parents of the commit correctly.

Question: **What are git hooks?**?

Git hooks are scripts that can run automatically on the occurrence of an event in a Git repository. These are used for automation of workflow in GIT. Git hooks also help in customizing the internal behavior of GIT. These are generally used for enforcing a GIT commit policy.

Question: What Are Disadvantages Of GIT?

GIT has very few disadvantages. These are the scenarios when GIT is difficult to use.

Some of these are:

Binary Files: If we have a lot binary files (non-text) in our project, then GIT becomes very slow. E.g. Projects with a lot of images or Word documents.

Steep Learning Curve: It takes some time for a newcomer to learn GIT. Some of the GIT commands are non-intuitive to a fresher.

Slow remote speed: Sometimes the use of remote repositories in slow due to network latency. Still GIT is better than other VCS in speed.

Question: What is stored inside a commit object in GIT?

GIT commit object contains following information:

SHA1 name: A 40 character string to identify a commit

Files: List of files that represent the state of a project at a specific point of time

Reference: Any reference to parent commit objects

Question: What Is GIT reset command?

Git reset command is used to reset current HEAD to a specific state. By default it reverses the action of git add command. So we use git reset command to undo the changes of git add command. Reference: Any reference to parent commit objects

Question: **How GIT protects the code in a repository**?

GIT is made very secure since it contains the source code of an organization. All the objects in a GIT repository are encrypted with a hashing algorithm called SHA1.

This algorithm is quite strong and fast. It protects source code and other contents of repository against the possible malicious attacks.

This algorithm also maintains the integrity of GIT repository by protecting the change history against accidental changes.

Continuos Integration Interview Questions

Question: What is Continuos Integration?

Continuous Integration is the process of continuously integrating the code and often multiple times per day. The purpose is to find problems quickly, s and deliver the fixes more rapidly.

CI is a best practice for software development. It is done to ensure that after every code change there is no issue in software.

Question: What Is Build Automation?

Build automation is the process of automating the creation of a software build and the associated processes.

Including compiling computer source code into binary code, packaging binary code, and running automated tests.

Question: What Is Automated Deployment?

Automated Deployment is the process of consistently pushing a product to various environments on a “trigger.”

It enables you to quickly learn what to expect every time you deploy an environment with much faster results.

This combined with Build Automation can save development teams a significant amount of hours.

Automated Deployment saves clients from being extensively offline during development and allows developers to build while “touching” fewer of a clients’ systems.

With an automated system, human error is prevented. In the event of human error, developers are able to catch it before live deployment – saving time and headache.

You can even automate the contingency plan and make the site rollback to a working or previous state as if nothing ever happened.

Clearly, this automated feature is super valuable in allowing applications and sites to continue during fixes.

Additionally, contingency plans can be version-controlled, improved and even self-tested.

Question: How Continuous Integration Implemented?

Different tools for supporting Continuous Integration are Hudson, Jenkins and Bamboo. Jenkins is the most popular one currently. They provide integration with various version control systems and build tools.

Question: How Continuous Integration process does work?

Whenever developer commits changes in version control system, then Continuous Integration server detects that changes are committed. And goes through following process

- Continuous Integration server retrieves latest copy of changes.

- It build code with new changes in build tools.
- If build fails notify to developer.
- After build pass run automated test cases if test cases fail notify to developer.
- Create package for deployment environment.

Question: What Are The Software Required For Continuous Integration process?

Here are the minimum tools you need to achieve CI

- Source code repository : To commit code and changes for example git.
- Server: It is Continuous Integration software for example Jenkin, Teamcity.

- Build tool: It builds application on particular way for example maven, gradle.
- Deployment environment : On which application will be deployed.

Question: **What Is Jenkins Software?**

Jenkins is self-contained, open source automation server used to automate all sorts of tasks related to building, testing, and delivering or deploying software.

Jenkins is one of the leading open source automation servers available. Jenkins has an extensible, plugin-based architecture, enabling developers to create 1,400+ plugins to adapt it to a multitude of build, test and deployment technology integrations.

Questions: What is a Jenkins Pipeline?

Jenkins Pipeline (or simply “Pipeline”) is a suite of plugins which supports implementing and integrating continuous delivery pipelines into Jenkins..

Question: **What is the difference between Maven, Ant,Gradle and Jenkins** ?

Maven and Ant are Build Technologies whereas Jenkins is a continuous integration tool.

Question: **Why do we use Jenkins?**

Jenkins is an **open-source** continuous integration software tool written in the Java programming language for testing and reporting on isolated changes in a larger code base in real time.

The **Jenkins software** enables developers to find and solve defects in a code base rapidly and to automate testing of their builds.

Question: **What are CI Tools?**?

Here is the list of the top 8 **Continuous Integration tools**:

- Jenkins
- TeamCity
- Travis CI
- Go CD
- Bamboo
- GitLab CI

- CircleCI
- Codeship

Question: **Which SCM tools Jenkins supports?**?

Jenkins supports version control tools, including AccuRev, CVS, Subversion, Git, Mercurial, Perforce, ClearCase and RTC, and can execute Apache Ant, Apache Maven and arbitrary shell scripts and Windows batch commands.

Question: **Why do we use Pipelines in Jenkins**?

Pipeline adds a powerful set of automation tools onto Jenkins, supporting use cases that span from simple continuous integration to comprehensive continuous delivery pipelines.

By modeling a series of related tasks, users can take advantage of the many features of

Pipeline:

- Code: Pipelines are implemented in code and typically checked into source control, giving teams the ability to edit, review, and iterate upon their delivery pipeline.
- Durable: Pipelines can survive both planned and unplanned restarts of the Jenkins master.
- Pausable: Pipelines can optionally stop and wait for human input or approval before continuing the Pipeline run.
- Versatile: Pipelines support complex real-world continuous delivery requirements, including the ability to fork/join, loop, and perform work in parallel.
- Extensible: The Pipeline plugin supports custom extensions to its DSL and multiple options for integration with other plugins.

Question: **How do you create Multibranch Pipeline in Jenkins**?

The Multi branch Pipeline project type enables you to implement different Jenkins files for different branches of the same project.

In a Multi branch Pipeline project, Jenkins automatically discovers, manages and executes Pipelines for branches which contain a Jenkins file in source control.

Question: **What are Jobs in Jenkins??**

Jenkins can be used to perform the typical build server work, such as doing continuous/official/nightly builds, run tests, or perform some repetitive batch tasks. This is called “**free-style software project**” in Jenkins.

Question: **How do you configuring automatic builds in Jenkins**?

Builds in Jenkins can be triggered periodically (on a schedule, specified in configuration), or when source changes in the project have been detected, or they can be automatically triggered by requesting the URL:

Question: **What is a Jenkins file?**

Jenkins file is a text file containing the definition of a Jenkins Pipeline and checks into source control.

Amazon AWS DevOps Interview Questions

Question: What is **Amazon Web Services**?

Amazon Web Services provides services that help you practice DevOps at your company and that are built first for use with AWS.

These tools automate manual tasks, help teams manage complex environments at scale, and keep engineers in control of the high velocity that is enabled by DevOps

Question: What Are Benefits Of AWS for DevOps?

There are many benefits of using AWS for devops

Get Started Fast: Each AWS service is ready to use if you have an AWS account. There is no setup required or software to install.

Fully Managed Services: These services can help you take advantage of AWS resources quicker. You can worry less about setting up, installing, and operating infrastructure on your own. This lets you focus on your core product.

Built For Scalability: You can manage a single instance or scale to thousands using AWS services. These services help you make the most of flexible compute resources by simplifying provisioning, configuration, and scaling.

Programmable: You have the option to use each service via the AWS Command Line Interface or through APIs and SDKs. You can also model and provision AWS resources and your entire AWS infrastructure using declarative AWS CloudFormation templates.

Automation: AWS helps you use automation so you can build faster and more efficiently. Using AWS services, you can automate manual tasks or processes such as deployments,

development & test workflows, container management, and configuration management.

Secure: Use AWS Identity and Access Management (IAM) to set user permissions and policies. This gives you granular control over who can access your resources and how they access those resources.

Question: **How To Handle Continuous Integration and Continuous Delivery in AWS** Devops?

The AWS Developer Tools help in securely store and version your application's source code and automatically build, test, and deploy your application to AWS.

Question: What Is The Importance Of Buffer In Amazon Web Services?

An Elastic Load Balancer ensures that the incoming traffic is distributed optimally across various AWS instances.

A buffer will synchronize different components and makes the arrangement additional elastic to a burst of load or traffic.

The components are prone to work in an unstable way of receiving and processing the requests.

The buffer creates the equilibrium linking various apparatus and crafts them effort at the identical rate to supply more rapid services.

Question: What Are The Components Involved In Amazon Web Services?

There are 4 components

Amazon S3 : with this, one can retrieve the key information which are occupied in creating cloud structural design and amount of produced information also can be stored in this component that is the consequence of the key specified.

Amazon EC2 instance : helpful to run a large distributed system on the Hadoop cluster.

Automatic parallelization and job scheduling can be achieved by this component.

Amazon SQS : this component acts as a mediator between different controllers. Also worn for cushioning requirements those are obtained by the manager of Amazon.

Amazon SimpleDB : helps in storing the transitional position log and the errands executed by the consumers.

Question: How is a Spot instance different from an On-Demand instance or Reserved Instance?

Spot Instance, On-Demand instance and Reserved Instances are all models for pricing. Moving along, spot instances provide the ability for customers to purchase compute capacity with no upfront commitment, at hourly rates usually lower than the On-Demand rate in each region.

Spot instances are just like bidding, the bidding price is called Spot Price. The Spot Price fluctuates based on supply and demand for instances, but customers will never pay more than the maximum price they have specified.

If the Spot Price moves higher than a customer's maximum price, the customer's EC2 instance will be shut down automatically.

But the reverse is not true, if the Spot prices come down again, your EC2 instance will not be launched automatically, one has to do that manually.

In Spot and On demand instance, there is no commitment for the duration from the user side, however in reserved instances one has to stick to the time period that he has chosen.

Questions: **What are the best practices for Security in Amazon EC2?**

There are several best practices to secure Amazon EC2. A few of them are given below:

- Use AWS Identity and Access Management (IAM) to control access to your AWS resources.
- Restrict access by only allowing trusted hosts or networks to access ports on your instance.
- Review the rules in your security groups regularly, and ensure that you apply the principle of least
- Privilege – only open up permissions that you require.

- Disable password-based logins for instances launched from your AMI. Passwords can be found or cracked, and are a security risk.

Question: **What is AWS CodeBuild in AWS** Devops**?**

AWS CodeBuild is a fully managed build service that compiles source code, runs tests, and produces software packages that are ready to deploy.

With CodeBuild, you don’t need to provision, manage, and scale your own build servers. CodeBuild scales continuously and processes multiple builds concurrently, so your builds are not left waiting in a queue.

Question: **What is Amazon Elastic Container Service in AWS** Devops**?**

Amazon Elastic Container Service (ECS) is a highly scalable, high performance container management service that supports Docker containers and allows you to easily run applications on a managed cluster of Amazon EC2 instances.

Question: **What is AWS Lambda in AWS Devops?**

AWS Lambda lets you run code without provisioning or managing servers. With Lambda, you can run code for virtually any type of application or backend service, all with zero administration.

Just upload your code and Lambda takes care of everything required to run and scale your code with high availability.

Splunk DevOps Interview Questions

Question: What is Splunk?

The platform of Splunk allows you to get visibility into machine data generated from different networks, servers, devices, and hardware.

It can give insights into the application management, threat visibility, compliance, security, etc. so it is used to analyze machine data. The data is collected from the forwarder from the source and forwarded to the indexer. The data is stored locally on a host machine or cloud. Then on the data stored in the indexer the search head searches, visualizes, analyzes and performs various other functions.

Question: What Are The Components Of Splunk?

The main components of Splunk are Forwarders, Indexers and Search Heads.Deployment Server(or Management Console Host) will come into the picture in case of a larger environment.

Deployment servers act like an antivirus policy server for setting up Exceptions and Groups so that you can map and create adifferent set of data collection policies each for either window based server or a Linux based server or a Solaris based server. **plunk has four important components :**

- **Indexer –** It indexes the machine data
- **Forwarder –** Refers to Splunk instances that forward data to the remote indexers
- **Search Head –** Provides GUI for searching
- **Deployment Server –**Manages the Splunk components like indexer, forwarder, and

search head in computing environment.

Question: **What are alerts in Splunk?**

An alert is an action that a saved search triggers on regular intervals set over a time range, based on the results of the search.

When the alerts are triggered, various actions occur consequently.. For instance, sending an email when a search to the predefined list of people is triggered.

Three types of alerts:

1. **Pre-result alerts :** Most commonly used alert type and runs in real-time for an all-time span. These alerts are designed such that whenever a search returns a result, they are triggered.
2. **Scheduled alerts :** The second most common- scheduled results are set up to evaluate the results of a historical search result running over a set time range on a regular schedule. You can define a time range, schedule and the trigger condition to an alert.
3. **Rolling-window alerts:** These are the hybrid of pre-result and scheduled alerts. Similar to the former, these are based on real-time search but do not trigger each time the search returns a matching result . It examines all events in real-time mapping within the rolling window and triggers the time that specific condition by that event in the window is met, like the scheduled alert is triggered on a scheduled search.

Question: What Are The Categories Of SPL Commands?

SPL commands are divided into five categories:

1. **Sorting Results –** Ordering results and (optionally) limiting the number of results.
2. **Filtering Results –** It takes a set of events or results and filters them into a smaller set of results.
3. **Grouping Results –** Grouping events so you can see patterns.

4. **Filtering, Modifying and Adding Fields –** Taking search results and generating a summary for reporting.
5. **Reporting Results –** Filtering out some fields to focus on the ones you need, or modifying or adding fields to enrich your results or events.

Question: **What Happens If The License Master Is Unreachable?**

In case the license master is unreachable, then it is just not possible to search the data.

However, the data coming in to the Indexer will not be affected. The data will continue to flow into your Splunk deployment.

The Indexers will continue to index the data as usual however, you will get a warning message on top your Search head or web UI saying that you have exceeded the indexing volume.

And you either need to reduce the amount of data coming in or you need to buy a higher capacity of license. Basically, the candidate is expected to answer that the indexing does not stop; only searching is halted

---

Question: What are common port numbers used by Splunk?

Common port numbers on which default services run are:

Service	**Port Number**
Splunk Management Port	8089
Splunk Index Replication Port	8080
KV store	8191
Splunk Web Port	8000
Splunk Indexing Port	9997
Splunk network port	514

Question: What Are Splunk Buckets? Explain The Bucket Lifecycle?

A directory that contains indexed data is known as a Splunk bucket. It also contains events of a certain period. Bucket lifecycle includes following stages:

- **Hot –** It contains newly indexed data and is open for writing. For each index, there are one or more hot buckets available
- **Warm –** Data rolled from hot
- **Cold –** Data rolled from warm
- **Frozen –** Data rolled from cold. The indexer deletes frozen data by default but users can also archive it.
- **Thawed –** Data restored from an archive. If you archive frozen data , you can later return it to the index by thawing (defrosting) it.

Question: **Explain Data Models and Pivot?**

Data models are used for creating a structured hierarchical model of data. It can be used when you have a large amount of unstructured data, and when you want to make use of that information without using complex search queries.

A few use cases of Data models are:

- **Create Sales Reports:** If you have a sales report, then you can easily create the total number of successful purchases, below that you can create a child object containing the list of failed purchases and other views
- **Set Access Levels:** If you want a structured view of users and their various access levels, you can use a data model

On the other hand with pivots, you have the flexibility to create the front views of your results and then pick and choose the most appropriate filter for a better view of results.

Question: What Is **File Precedence In Splunk?**

File precedence is an important aspect of troubleshooting in Splunk for an administrator, developer, as well as an architect.

All of Splunk's configurations are written in .conf files. There can be multiple copies present for each of these files, and thus it is important to know the role these files play when a Splunk instance is running or restarted. To determine the priority among copies of a configuration file, Splunk software first determines the directory scheme. The directory schemes are either a) Global or b) App/user. When the context is global (that is, where there's no app/user context), directory priority descends in this order:

1. System local directory — *highest priority*
2. App local directories
3. App default directories
4. System default directory — *lowest priority*

When the context is app/user, directory priority descends from user to app to system:

1. User directories for current user — *highest priority*
2. App directories for currently running app (local, followed by default)

3. App directories for all other apps (local, followed by default) — for exported settings only
4. System directories (local, followed by default) — *lowest priority*

Question: D**ifference Between Search Time And Index Time Field Extractions?**

Search time field extraction refers to the fields extracted while performing searches.

Whereas, fields extracted when the data comes to the indexer are referred to as Index time field extraction.

You can set up the indexer time field extraction either at the forwarder level or at the indexer level.

Another difference is that Search time field extraction's extracted fields are not part of the metadata, so they do not consume disk space.

Whereas index time field extraction's extracted fields are a part of metadata and hence consume disk space.

Question: **What Is Source Type In Splunk?**

Source type is a default field which is used to identify the data structure of an incoming event. Source type determines how Splunk Enterprise formats the data during the indexing process.

Source type can be set at the forwarder level for indexer extraction to identify different data formats.

Question: What is SOS?

SOS stands for Splunk on Splunk. It is a Splunk app that provides graphical view of your Splunk environment performance and issues.

It has following purposes:

- Diagnostic tool to analyze and troubleshoot problems
- Examine Splunk environment performance
- Solve indexing performance issues
- Observe scheduler activities and issues
- See the details of scheduler and user driven search activity
- Search, view and compare configuration files of Splunk

Question: What Is Splunk Indexer And Explain Its Stages?

The indexer is a Splunk Enterprise component that creates and manages indexes.
The main functions of an indexer are:

- Indexing incoming data
- Searching indexed data **Splunk indexer has following stages:**

Input : Splunk Enterprise acquires the raw data from various input sources and breaks it into 64K blocks and assign them some metadata keys. These keys include host, source and source type of the data. **Parsing :** Also known as event processing, during this stage, the Enterprise analyzes and transforms the data, breaks data into streams, identifies, parses and sets timestamps, performs metadata annotation and transformation of data. **Indexing :** In this phase, the parsed events are written on the disk index including both compressed data and the associated index files. **Searching :** The ‘Search’ function plays a

major role during this phase as it handles all searching aspects (interactive, scheduled searches, reports, dashboards, alerts) on the indexed data and stores saved searches, events, field extractions and views

Question: State The Difference Between Stats and Eventstats Commands?

Stats – This command produces summary statistics of all existing fields in your search results and store them as values in new fields. **Eventstats –** It is same as stats command except that aggregation results are added in order to every event and only if the aggregation is applicable to that event. It computes the requested statistics similar to stats but aggregates them to the original raw data.

log4J DevOps Interview Questions

Question: What is log4j?

log4j is a reliable, fast and flexible logging framework (APIs) written in Java, which is distributed under the Apache Software License.

log4j has been ported to the C, C++, C#, Perl, Python, Ruby, and Eiffel languages.

log4j is highly configurable through external configuration files at runtime. It views the logging process in terms of levels of priorities and offers mechanisms to direct logging information to a great variety of destinations.

Question: **What Are The Features Of Log4j**

Log4j is widely used framework and here are features of log4j

- It is thread-safe.It is optimized for speed
- It is based on a named logger hierarchy.
- It supports multiple output appenders per logger.
- It supports internationalization.
- It is not restricted to a predefined set of facilities.
- Logging behavior can be set at runtime using a configuration file.
- It is designed to handle Java Exceptions from the start.
- It uses multiple levels, namely ALL, TRACE, DEBUG, INFO, WARN, ERROR and FATAL.
- The format of the log output can be easily changed by extending the Layout class.
- The target of the log output as well as the writing strategy can be altered by implementations of the Appender interface.
- It is fail-stop. However, although it certainly strives to ensure delivery, log4j does not

guarantee that each log statement will be delivered to its destination.

Question: What are the components of log4j**?**

log4j has three main components

- loggers: Responsible for capturing logging information.
- appenders: Responsible for publishing logging information to various preferred destinations.
- layouts: Responsible for formatting logging information in different styles.

Question: How do you initialize and use Log4J ?

public class LoggerTest { static Logger log = Logger.getLogger (LoggerTest.class.getName()); public void my logerMethod() { if(log.isDebugEnabled()) log.debug("This is test message" + var2);) } }

Question: **What are Pros and Cons of Logging?**

Following are the Pros and Cons of Logging Logging is an important component of the software development. A well-written logging code offers quick debugging, easy maintenance, and structured storage of an application's runtime information. Logging does have its drawbacks also. It can slow down an application. If too verbose, it can cause scrolling blindness. To alleviate these concerns, log4j is designed to be reliable, fast and extensible. Since logging is rarely the main focus of an application, the log4j API strives to be simple to understand and to use.

Question:**What Is The Purpose Of Logger Object?**

Logger Object − The top-level layer of log4j architecture is the Logger which provides the Logger object.

The Logger object is responsible for capturing logging information and they are stored in a namespace hierarchy.

Question: What is the purpose of Layout object?

The layout layer of log4j architecture provides objects which are used to format logging information in different styles. It provides support to appender objects before publishing logging information.

Layout objects play an important role in publishing logging information in a way that is human-readable and reusable.

Questions: What is the purpose of Appender object?

The Appender object is responsible for publishing logging information to various preferred destinations such as a database, file, console, UNIX Syslog, etc.

Question: **What Is The Purpose Of ObjectRenderer Object?**

The ObjectRenderer object is specialized in providing a String representation of different objects passed to the logging framework.

This object is used by Layout objects to prepare the final logging information.

Question: **What Is LogManager object?**

The LogManager object manages the logging framework. It is responsible for reading the initial configuration parameters from a system-wide configuration file or a configuration class.

Question: **How Will You Define A File Appender Using Log4j.properties?**

Following syntax defines a file appender −
log4j.appender.FILE=org.apache.log4j.FileAppender
log4j.appender.FILE.File=${log}/log.out

Question: W**hat Is The Purpose Of Threshold In Appender?**

Appender can have a threshold level associated with it independent of the logger level. The Appender ignores any logging messages that have a level lower than the threshold level.

Docker DevOps Interview Questions

Question: What is Docker?

Docker provides a container for managing software workloads on shared infrastructure, all while keeping them isolated from one another.

Docker is a tool designed to make it easier to create, deploy, and run applications by using containers.

Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package.

By doing so, the developer can rest assured that the application will run on any other Linux machine regardless of any customized settings that machine might have that could differ from the machine used for writing and testing the code. In a way, Docker is a bit like a virtual machine. But unlike a virtual machine, rather than creating a whole virtual operating system. Docker allows applications to use the same Linux kernel as the system that they're running on and only requires applications be shipped with things not already running on the host computer. This gives a significant performance boost and reduces the size of the application.

Question: **What Are Linux Containers**?

Linux containers, in short, contain applications in a way that keep them isolated from the host system that they run on.

Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package.

And they are designed to make it easier to provide a consistent experience as developers and system administrators move code from development environments into production in a fast and replicable way.

Question: Who Is Docker For**?**

Docker is a tool that is designed to benefit both developers and system administrators, making it a part of many DevOps (developers + operations) toolchains.

For developers, it means that they can focus on writing code without worrying about the system that it will ultimately be running on.

It also allows them to get a head start by using one of thousands of programs already designed to run in a Docker container as a part of their application.

For operations staff, Docker gives flexibility and potentially reduces the number of systems needed because of its small footprint and lower overhead.

Question: **What Is Docker Container?**

Docker containers include the application and all of its dependencies, but share the kernel with other containers, running as isolated processes in user space on the host operating system.

Docker containers are not tied to any specific infrastructure: they run on any computer, on any infrastructure, and in any cloud.

Now explain how to create a Docker container, Docker containers can be created by either creating a Docker image and then running it or you can use Docker images that are present on the Dockerhub. Docker containers are basically runtime instances of Docker images.

Question: **What Is Docker Image?**

Docker image is the source of Docker container. In other words, Docker images are used to create containers.

Images are created with the build command, and they'll produce a container when started with run.

Images are stored in a Docker registry such as registry.hub.docker.com because they can become quite large, images are designed to be composed of layers of other images, allowing a minimal amount of data to be sent when transferring images over the network.

Question: **What Is Docker Hub?**

Docker hub is a cloud-based registry service which allows you to link to code repositories, build your images and test them, stores manually pushed images, and links to Docker cloud so you can deploy images to your hosts.

It provides a centralized resource for container image discovery, distribution and change management, user and team collaboration, and workflow automation throughout the development pipeline.

Question: What is Docker Swarm?

Docker Swarm is native clustering for Docker. It turns a pool of Docker hosts into a single, virtual Docker host.

Docker Swarm serves the standard Docker API, any tool that already communicates with a Docker daemon can use Swarm to transparently scale to multiple hosts.

I will also suggest you to include some supported tools:

- Dokku
- Docker Compose
- Docker Machine
- Jenkins

Questions: **What is Dockerfile used for**?

A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image.

Using docker build users can create an automated build that executes several command-line instructions in succession.

Question: **How is Docker different from other container technologies?**

Docker containers are easy to deploy in a cloud. It can get more applications running on the same hardware than other technologies.

It makes it easy for developers to quickly create, ready-to-run containerized applications and it makes managing and deploying applications much easier. You can even share containers with your applications.

Question: **How to create Docker container?**

We can use Docker image to create Docker container by using the below command:

```
1 docker run -t -i command name
```

This command will create and start a container. You should also add, If you want to check the list of all running container with the status on a host use the below command:

```
1 docker ps - a
```

Question: **How to stop and restart the Docker container?**

In order to stop the Docker container you can use the below command:

```
1 docker stop container
  ID
```

Now to restart the Docker container you can use:

```
1  docker restart container
   ID
```

Question: What is the difference between docker run and docker create?

The primary difference is that using **‘docker create’** creates a container in a stopped state. **Bonus point:** You can use **‘docker create’** and store an outputed container ID for later use. The best way to do it is to use **‘docker run’** with -**-cidfile FILE_NAME** as running it again won’t allow to overwrite the file.

Question: What four states a Docker container can be in?

- Running
- Paused
- Restarting
- Exited

Question:What Is Difference Between Repository and a Registry?

Docker registry is a service for hosting and distributing images. Docker repository is a collection of related Docker images.

Question: How to link containers?

The simplest way is to use network port mapping. There's also the **- -link** flag which is deprecated.

Question: What is the difference between Docker RUN, CMD and ENTRYPOINT?

A **CMD** does not execute anything at build time, but specifies the intended command for the image.

RUN actually runs a command and commits the result.

If you would like your container to run the same executable every time, then you should consider using **ENTRYPOINT** in combination with CMD.

Question: How many containers can run per host?

As far as the number of containers that can be run, this really depends on your

environment. The size of your applications as well as the amount of available resources will all affect the number of containers that can be run in your environment.

Containers unfortunately are not magical. They can’t create new CPU from scratch. They do, however, provide a more efficient way of utilizing your resources.

The containers themselves are super lightweight (remember, shared OS vs individual OS per container) and only last as long as the process they are running. Immutable infrastructure if you will.

Question: **What is Docker hub?**

Docker hub is a cloud-based registry service which allows you to link to code repositories, build your images and test them, stores manually pushed images, and links to Docker cloud so you can deploy images to your hosts.

It provides a centralized resource for container image discovery, distribution and change management, user and team collaboration, and workflow automation throughout the development pipeline.

VmWare DevOps Interview Questions

Question: What is VmWare?

VMware was founded in 1998 by five different IT experts. The company officially launched its first product, VMware Workstation, in 1999, which was followed by the VMware GSX Server in 2001. The company has launched many additional products since that time. VMware's desktop software is compatible with all major OSs, including Linux, Microsoft Windows, and Mac OS X. VMware provides three different types of desktop software:

- VMware Workstation: This application is used to install and run multiple copies or instances of the same operating systems or different operating systems on a single physical computer machine.
- VMware Fusion: This product was designed for Mac users and provides extra compatibility with all other VMware products and applications.

- VMware Player: This product was launched as freeware by VMware for users who do not have licensed VMWare products. This product is intended only for personel use.

VMware's software hypervisors intended for servers are bare-metal embedded hypervisors that can run directly on the server hardware without the need of an extra primary OS. VMware’s line of server software includes:

- VMware ESX Server: This is an enterprise-level solution, which is built to provide better functionality in comparison to the freeware VMware Server resulting from a lesser system overhead. VMware ESX is integrated with VMware vCenter that provides additional solutions to improve the manageability and consistency of the server implementation.

- VMware ESXi Server: This server is similar to the ESX Server except that the service

console is replaced with BusyBox installation and it requires very low disk space to operate.

- VMware Server: Freeware software that can be used over existing operating systems like Linux or Microsoft Windows.

Question: **What is Virtualization?**

The process of creating virtual versions of physical components i-e Servers, Storage Devices, Network Devices on a physical host is called virtualization.

Virtualization lets you run multiple virtual machines on a single physical machine which is called ESXi host.

Question: **What are different types of virtualization?**

There are 5 basic types of virtualization

- Server virtualization: consolidates the physical server and multiple OS can be run on a single server.
- Network Virtualization: Provides complete reproduction of physical network into a software defined network.
- Storage Virtualization: Provides an abstraction layer for physical storage resources to manage and optimize in virtual deployment.
- Application Virtualization: increased mobility of applications and allows migration of VMs from host on another with minimal downtime.
- Desktop Virtualization: virtualize desktop to reduce cost and increase service

---

Question: **What is Service Console?**

The service console is developed based up on Redhat Linux Operating system, it is used to manage the VMKernel

Question: **What is vCenter Agent?**

VC agent is an agent installed on ESX server which enables communication between VC and ESX server.

This Agent will be installed on ESX/ESXi will be done when you try to add the ESx host in Vcenter.

Question: **What is VMKernel?**

VMWare Kernel is a Proprietary kernel of vmware and is not based on any of the flavors of Linux operating systems.

VMkernel requires an operating system to boot and manage the kernel. A service console is being provided when VMWare kernel is booted.

Only service console is based up on Redhat Linux OS not VMkernel.

Question:**What is VMKernel and why it is important?**

VMkernel is a virtualization interface between a Virtual Machine and the ESXi host which stores VMs.

It is responsible to allocate all available resources of ESXi host to VMs such as memory, CPU, storage etc.

It’s also control special services such as vMotion, Fault tolerance, NFS, traffic management and iSCSI.

To access these services, VMkernel port can be configured on ESXi server using a standard or distributed vSwitch. Without VMkernel, hosted VMs cannot communicate with ESXi server.

Question: **What is hypervisor and its types?**

Hypervisor is a virtualization layer that enables multiple operating systems to share a single hardware host.

Each operating system or VM is allocated physical resources such as memory, CPU, storage etc by the host. There are two types of hypervisors

- Hosted hypervisor (works as application i-e VMware Workstation) Bare-metal (is
- virtualization software i-e VMvisor, hyper-V which is installed directly onto the hardware and controls all physical resources).

Questions: **What is virtual networking?**

A network of VMs running on a physical server that are connected logically with each other is called virtual networking.

Question: **What is vSS?**

---

vSS stands for Virtual Standard Switch is responsible for communication of VMs hosted on a single physical host.

it works like a physical switch automatically detects a VM which want to communicate with other VM on a same physical server.

Question: **What is VMKernal adapter and why it used?**

AVMKernel adapter provides network connectivity to the ESXi host to handle network traffic for vMotion, IP Storage, NAS, Fault Tolerance, and vSAN.

For each type of traffic such as vMotion, vSAN etc. separate VMKernal adapter should be created and configured.

Question:**What are three port groups are configured in ESXi networking?**

- Virtual Machine Port Group – Used for Virtual Machine Network
- Service Console Port Group – Used for Service Console Communications
- VMKernel Port Group – Used for VMotion, iSCSI, NFS Communications

Question: W**hat are main components of vCenter Server architecture?**

There are three main components of vCenter Server architecture.

- vSphere Client and Web Client: a user interface.
- vCenter Server database: SQL server or embedded PostgreSQL to store inventory, security roles, resource pools etc.
- SSO: a security domain in virtual environment

Question: **What is datastore?**

A Datastore is a storage location where virtual machine files are stored and accessed. Datastore is based on a file system which is called VMFS, NFS

Question: **How many disk types are in VMware?**

There are three disk types in vSphere.

1. Thick Provisioned Lazy Zeroes: every virtual disk is created by default in this disk format. Physical space is allocated to a VM when virtual disk is created. It can’t be converted to thin disk.
2. Thick Provision Eager Zeroes: this disk type is used in VMware Fault Tolerance. All required disk space is allocated to a VM at time of creation. It takes more time to create a virtual disk compare to other disk formats.

3. Thin provision: It provides on-demand allocation of disk space to a VM. When data size grows, the size of disk will grow. Storage capacity utilization can be up to 100% with thin provisioning.
4. What is Storage vMotion?

It is similar to traditional vMotion, in Storage vMotion, virtual disk of a VM is moved from datastore to another. During Storage vMotion, virtual disk types think provisioning disk can be transformed to thin provisioned disk.

Question: **What is the use of VMKernel Port ?**

Vmkernel port is used by ESX/ESXi for vmotion, ISCSI & NFS communications. ESXi uses Vmkernel as the management network since it don't have serviceconsole built with it.

Question: **What are different types of Partitions in ESX server?**

AC/-root Swap /var /Var/core /opt /home /tmp

Question: E**xplain What Is VMware DRS?**

VMware DRS stands for Distributed Resource Scheduler; it dynamically balances resources across various host under cluster or resource pool. It enables users to determine the rules and policies which decide how virtual machines deploy resources, and these resources should be prioritized to multiple virtual machines.

DevOps Testing Interview Questions

Question: **What is Continuous Testing?**

Continuous Testing is the process of executing automated tests to obtain immediate feedback on the business risks associated with in the latest build.

In this way, each build is tested continuously, allowing Development teams to get fast feedback so that they can prevent those problems from progressing to the next stage of Software delivery life-cycle.

Question: **What is Automation Testing**

Automation testing is a process of automating the manual testing process. Automation testing involves use of separate testing tools, which can be executed repeatedly and

doesn’t require any manual intervention.

Question: **What Are The Benefits of Automation Testing?**

Here are some of the benefits of using Continuous Testing;

- Supports execution of repeated test cases
- Aids in testing a large test matrix
- Enables parallel execution
- Encourages unattended execution
- Improves accuracy thereby reducing human generated errors
- Saves time and money

Question: **Why is Continuous Testing important for DevOps?**

Continuous Testing allows any change made in the code to be tested immediately.

This avoids the problems created by having “big-bang” testing left to the end of the development cycle such as release delays and quality issues.

In this way, Continuous Testing facilitates more frequent and good quality releases.”

Question: **What are the Testing types supported by Selenium?**

Selenium supports two types of testing:

Regression Testing: It is the act of retesting a product around an area where a bug was fixed.

Functional Testing: It refers to the testing of software features (functional points) individually.

Question: **What is the Difference Between Assert and Verify commands in Selenium?**

Assert command checks whether the given condition is true or false.

Verify command also checks whether the given condition is true or false. Irrespective of the condition being true or false, the program execution doesn’t halts i.e. any failure during verification would not stop the execution and all the test steps would be executed.

Summary

DevOps refers to a wide range of tools, process and practices used bycompanies to improve their build, deployment, testing and release life cycles.

In order to ace a DevOps interview you need to have a deep understanding of all of these tools and processes.

Most of the technologies and process used to implement DevOps are not isolated. Most probably you are already familiar with many of these. All you have to do is to prepare for these from DevOps perspective.

In this guide I have created the largest set of interview questions. Each section in this guide caters to a specific area of DevOps.

In order to increase your chances of success in DevOps interview you need to go through all of these questions.

References

https://theagileadmin.com/what-is-devops/

https://en.wikipedia.org/wiki/DevOps

http://www.javainuse.com/misc/gradle-interview-questions

https://mindmajix.com/gradle-interview-questions

https://tekslate.com/groovy-interview-questions-and-answers/

https://mindmajix.com/groovy-interview-questions

https://www.wisdomjobs.com/e-university/groovy-programming-language-interviewquestions.html

https://www.quora.com/What-are-some-advantages-of-the-Groovy-programminglanguage

70/71

https://www.quora.com/What-are-some-advantages-of-the-Groovy-programminglanguage

http://groovy-lang.org/documentation.html

https://maven.apache.org/guides/introduction/introduction-to-archetypes.html

https://en.wikipedia.org/wiki/Apache_Maven

https://www.tecmint.com/linux-process-management/

https://www.tecmint.com/dstat-monitor-linux-server-performance-process-memorynetwork/

https://www.careerride.com/Linux-Interview-Questions.aspx

https://www.onlineinterviewquestions.com/git-interview-questions/#.WxcTP9WFMy4

https://www.atlassian.com/git/tutorials/what-is-git

https://www.toptal.com/git/interview-questions

https://www.sbf5.com/~cduan/technical/git/git-1.shtml

http://preparationforinterview.com/preparationforinterview/continuous-integrationinterview-question

https://codingcompiler.com/jenkins-interview-questions-answers/

https://www.edureka.co/blog/interview-questions/top-splunk-interview-questions-andanswers/

https://intellipaat.com/interview-question/splunk-interview-questions/

https://www.edureka.co/blog/interview-questions/docker-interview-questions/

http://www.vmwarearena.com/vmware-interview-questions-and-answers/

https://www.myvirtualjourney.com/top-80-vmware-interview-questions-answers/

https://www.edureka.co/blog/interview-questions/top-devops-interview-questions2016/