

Devops

1) Explain what is DevOps?

It is a newly emerging term in IT field, which is nothing but a practice that emphasizes the collaboration and communication of both software developers and other information-technology (IT) professionals. It focuses on delivering software product faster and lowering the failure rate of releases.

2) Mention what are the key aspects or principle behind DevOps?

The key aspects or principle behind DevOps is

Infrastructure as code

Continuous deployment

Automation

Monitoring

Security

3) What are the core operations of DevOps with application development and with infrastructure?

The core operations of DevOps with

Application development

Code building

Code coverage

Unit testing

Packaging

Deployment

4) What is GIT?

GIT is a distributed version control system and source code management (SCM) system with an emphasis to handle small and large projects with speed and efficiency.

5) What is a repository in GIT?

A repository contains a directory named .git, where git keeps all of its metadata for the repository. The content of the .git directory are private to git.

6) What is the difference between GIT and SVN?

The difference between GIT and SVN is

- a) Git is less preferred for handling extremely large files or frequently changing binary files while SVN can handle multiple projects stored in the same repository.
- b) GIT does not support 'commits' across multiple branches or tags. Subversion allows the creation of folders at any location in the repository layout.
- c) Gits are unchangeable, while Subversion allows committers to treat a tag as a branch and to create multiple revisions under a tag root.

7) What are the advantages of using GIT?

- a) Data redundancy and replication
- b) High availability
- c) Only one .git directory per repository

- d) Superior disk utilization and network performance
- e) Collaboration friendly
- f) Any sort of projects can use GIT

8) Why GIT better than Subversion?

GIT is an open source version control system; it will allow you to run 'versions' of a project, which show the changes that were made to the code overtime also it allows you keep the backtrack if necessary and undo those changes. Multiple developers can checkout, and upload changes and each change can then be attributed to a specific developer.

9) What is "Staging Area" or "Index" in GIT?

Before completing the commits, it can be formatted and reviewed in an intermediate area known as 'Staging Area' or 'Index'.

10) What is GIT stash?

GIT stash takes the current state of the working directory and index and puts in on the stack for later and gives you back a clean working directory. So in case if you are in the middle of something and need to jump over to the other job, and at the same time you don't want to lose your current edits then you can use GIT stash.

11) What is the function of git clone?

The git clone command creates a copy of an existing Git repository. To get the copy of a central repository, 'cloning' is the most common way used by programmers.

12) What does commit object contain?

- a) A set of files, representing the state of a project at a given point of time
- b) Reference to parent commit objects
- c) An SHA1 name, a 40 character string that uniquely identifies the commit object.

13) How can you create a repository in Git?

In Git, to create a repository, create a directory for the project if it does not exist, and then run command "git init". By running this command .git directory will be created in the project directory, the directory does not need to be empty.

14) What is the purpose of branching in GIT?

The purpose of branching in GIT is that you can create your own branch and jump between those branches. It will allow you to go to your previous work keeping your recent work intact.

15) What is the common branching pattern in GIT?

The common way of creating branch in GIT is to maintain one as “Main” branch and create another branch to implement new features. This pattern is particularly useful when there are multiple developers working on a single project.

16) What is a ‘conflict’ in git?

A ‘conflict’ arises when the commit that has to be merged has some change in one place, and the current commit also has a change at the same place. Git will not be able to predict which change should take precedence.

17) What is GIT version control?

With the help of GIT version control, you can track the history of a collection of files and includes the functionality to revert the collection of files to another version. Each version captures a snapshot of the file system at a certain point of time. A collection of files and their complete history are stored in a repository.

18) What is Chef?

Chef is a powerful configuration management tool and automation tool. It converts the infrastructure of the company into a structured format of code. Thanks to Chef you can develop scripts that can be used for automating the IT and business process.

19) Explain the major components of Chef?

The architecture of Chef can be broken down into the Chef Server, Chef Workstation and Chef Node.

Chef Server – You can think of the Chef Server as the central store for that accumulates all the data that is necessary for configuring the nodes.

Chef Node – The Chef Node can be thought of as a client responsible for sharing data across network and it is based on the chef-client architecture.

Chef Workstation – The Chef Workstation can be thought of as the host for modifying the configuration data and cookbooks which is then forwarded to the Chef Server.

20) What is a Chef resource and what are its functions?

You can think of the Chef resource as a part of the infrastructure which is used for installing or running a service. Here we will discuss some of the important Chef resources:

The desired state of a configuration item can be described

To bring a certain item to a state you will know what are the steps involved

You can choose the resources like package, template or services

You can list the resource properties and other details needed

You can group the resources into recipes for describing working configuration.

21) What are Chef Resources and its functions?

When you group resources, it gets converted into a Recipe and this describes the working configuration and the policy. Using the Recipe, you can get all that is necessary for configuring a specific system. So here we list some of the important functions of a Recipe:

With the Chef Recipe you can install the software components

You can manage the files and apps deployment

You can execute the other recipes using one recipe.

22) What is a Chef Node and what is its importance?

We can think of the node as a physical server or a virtual machine that is a constituent of the Chef architecture. You can execute any resource with Chef.

23) What is the difference between a Cookbook and a Recipe in Chef?

When you group resources together what you get is a Recipe and this is useful in executing the configurations and policy. When you combine Recipes what you get is a Cookbook and this is easily manageable as compared to a Recipe.

24) What is the difference between a Cookbook and a Recipe in Chef?

When the action for a Chef Resource is not defined then it will automatically choose the Default action.

25) How does a Chef Repository work?

You can think of the Chef Repository as a collection of Cookbooks, roles, environments, data bags and more. It is possible to sync the Chef Repository with the Git and a Version Control System in order to improve the performance of it.

26) Explain the run-list in Chef?

The run-list is needed for specifying the Recipes that are needed for running it and the order in which the Recipes will be executed.

Some of the advantages of run-list include:

It ensures that the Recipes are running in the same order as specified

The node on which the run-list is to be executed has to be specified

It is transferred from the Workstation to Chef Server and the management console.

27) What is the importance of Chef starter kit?

The starter kit is needed for creating the configuration files in Chef. It gives you the information for interacting with the server and defining the configuration file. You can easily download the starter kit and use it at the desired place on the workstation.

28) What is the process for updating a Chef Cookbook?

we give you the steps to follow for updating a Chef Cookbook:

From the Workstation run the Knife SSH

Run the SSH and Chef-client on the server directly

You can use the Chef-client as a daemon for restarting the service as needed.

29) What is the process for bootstrapping in Chef and the information needed?

If you want to bootstrap in Chef, then you need the following information:

The hostname or Public IP address of the node

The user name and password for logging into a particular node

Using keys as authentication rather than any login credentials.

30) What is Docker?

You can define Docker as a containerization platform that combines all your applications in a package so that you have all the dependencies to run your applications in any environment. This means your application will run seamlessly on any environment and this makes it easy for having a product ready application. What Docker does is wrap the software that is needed in a file system that has everything for running the code, providing the runtime and all the necessary libraries and system tools. Containerization technology like Docker will share the same operating system kernel with the machine and due to this it is

extremely fast. This means that you have to run the Docker only at the beginning and after that since your OS is already running, you will have a smooth and seamless process.

31) What is the benefit of using a Docker over a hypervisor?

Though Docker and Hypervisor might do the same job overall there are many differences between them in terms of how they work. Docker can be thought of as light weight since it uses very less resources and also the host kernel rather than creating it like a Hypervisor.

32) What are the unique features of Docker over other containerization technology?

we list some of the most important and unique features of Docker that makes it a top containerization technology unlike any other in the market today

You can run your Docker container either on your PC or your enterprise IT system

Along with the Docker Hub which is a repository of all containers you can deploy and download all your applications from a central location

You can even share your applications with the containers that you create.

33) What is Docker image?

Here we will be explaining what is the Docker image. The Docker image help to create the Docker containers. You can create the Docker image with the build command, due to this it creates a container that starts when it begins to run. All the docker images are stored in the Docker registry like the public docker registry. These have minimal amounts of layers within the image so that there is minimum amount of data on the network.

34) What is Docker container?

Here we will be discussing what is a Docker container. It is a comprehensive set of applications including all its dependencies which share the same OS kernel along with the other containers running in separate processes within the operating system in a user space. The Docker is not tied to any IT infrastructure and thus it can run on any computer system or the cloud. You can create a Docker container using the Docker images and then running it or you can use the images that are already created in the Docker Hub. To simplify things, let us say that the Docker containers are just runtime instances of the Docker image.

35) What is Docker hub?

You can think of Docker Hub as a cloud registry that lets you link the code repositories, create the images and test them. You can also store your pushed images, or you can link to the Docker Cloud, so that the images can be deployed to the host. You have a centralized container image discovery resource which can be used for collaboration of your teams, automating the workflow, distribution and change management by creating the development pipeline.

36) What is the use of Dockerfile?

The Dockerfile can be thought of as a set of instructions that you need to pass on to the Docker so that the images can be built from the specified instructions in the Dockerfile. You can think of the Dockerfile as a text document which has all the commands that are needed for creating a Docker image. You can create an automated build that lets you execute multiple command-lines one after the other.

37) What is the process for creating a Docker container?

You can use any of the specific Docker image for creating a Docker container using the below command.

```
docker run -t -i command name
```

This command not only creates the container but also will start it for you. If you want to check if the Docker container has been created or not then you need to have the following command which will list all the Docker containers along with the host on which the Docker container runs.

```
docker ps -a
```

38) What is Ansible?

Ansible is an open source automation platform which can help you with configuration management, task automation and application deployment. Ansible uses SSH installed on all systems unlike other configuration softwares that work on agent architecture. Ansible also do IT orchestration where you run tasks and create a chain of events

that happen on different servers and devices. It is written on Python language which needs to be installed on the remote host. Ansible is very easy to set up yet it is a very powerful tool for software deployment.

39) Talk about Ansible architecture.

Ansible works on 'agentless architecture'. It works by connecting to your nodes and pushes out Ansible modules to them which are small programmes. These modules are written to the resource nodes of the desired state of the system. With the help of SSH, Ansible then executes these modules and removes them when done. Since ansibles are based on agentless architecture, your pool of modules can dwell on any machine without requiring any server, daemons or databases. You just require terminal program, text editor and a version control system to keep a check on changes to your content. Ansible's "authorized_key" is used to give directions as to what machines will use which hosts.

40) Give a comparison between Ansible and puppet.

A comparative study between Ansible and puppet is given below:

Ansible: Ansible is very simple to set up. It is a simple technology which is written in YAML language. It is based on agent-less architecture which doesn't require nodes to locally install daemons. It facilitates automated workflow for continuous and hassle free delivery. Ansible doesn't support windows. It comes with good GUI and CLI accepts command in almost every language.

Puppet: puppet is a complex technology as compared with Ansible. It is written in Ruby language. It works on easy installation and facilitates visualisation and reporting. It is not based on agentless architecture and unlike Ansible, puppet supports for almost all major operating systems. The prerequisite for using puppet is that the user must learn the puppet DSL language.

41) How to keep secret data in playbook?

If you want to keep secret data in your ansible content and still share it publicly, then you can use Vault in playbooks.

42) When should you test playbooks and roles?

In Ansible, tests are added either in new playbooks or to the existing ones. So, most of the testing job offers clear hosting each time and with this method, you need to make very minor changes to coding.

43) What is Ansible role?

The very first step in creating an Ansible role is creating its directory structure.

44) List some advantages of using Ansible.

Unlike other configuration management system, Ansible is the most sought after software applications these days. it offers the following benefits to its users:

Agentless- Its work structure makes use of agentless architecture. The nodes are not required to install and run background daemons to connect with a controlling machine.

Low overhead- due to agentless model, Ansibles reduces the overheads on the network by preventing the nodes from polling the controlling machine.

Secure and consistent- Ansible only uses SSH and Python on the managed nodes. This ensures safety and security. Also, Ansible ensures consistent environments.

Reliable- an Ansible playbook can be idempotent when written carefully. This prevents unexpected side-effects on the managed systems.

Good performance- Ansible delivers flawless performance. Though it is very easy to set up yet it is a powerful tool for deploying software applications using SSH.

45) What are the software prerequisites that must be met before Jenkins is installed?

The software prerequisites for installing Jenkins is that first you need to install Java Development Kit. It also needs you to install the Jakarta Enterprise Edition. Jenkins also comes with an embedded Jetty runtime that can be used if WebSphere or Tomcat is not available.

46) How to configure and use third-party tools in Jenkins?

These are some of the steps used for working with a third-party tool in Jenkins.

You have to first install the third-party software

You need to have the plug-in that supports the third-party tool.

You have to configure the third-party tool in the admin console.

You can then use the plug-in from the Jenkins build job.

47) How to take a backup of your Jenkins build jobs?

Within the XML configuration each Jenkins build is stored. When this folder is copied, the configuration of all the build jobs that are managed by the Jenkins master are backed up. If you can perform a Jenkins Git integration, then it is good. When you copy the contents of the folder, you will see that the build jobs described in the folder will be restored when the Jenkins server is started the next time.

48) what are the steps included in a Jenkins pipeline.

A complete Jenkins pipeline will include building a project from the source code, putting it through a variety of unit, integrating, testing for user acceptance and performance and then finally deploying the packaged application on an application server.

So the steps in a Jenkins pipeline will include:

Build

Test

Deploy

49) Explain what is the Jenkins tool?

Jenkins can be thought of as an open source automation tool that is used for continuous integration. You will be able to continuously test your software projects so that the developers will be able to integrate the changes to the project. You can also integrate with a large number of testing and deployment technologies.

50) State some of the advantages of using Jenkins?

Here are some of the most important advantages of Jenkins:

You will get an automated build report every time a change is made to the source code

You will be able to achieve continuous integration with agile methodology principles

You can automate the maven release project with a few simple steps

The bugs can be easily tracked at the early development stage.

51) What are the requirements for using Jenkins?

Here we list some of the requirements for using Jenkins:

A source code repository like a Git repository

A build script like a Maven script that is checked into the repository.

52) How to schedule builds in Jenkins?

Here there are some steps for scheduling of builds in Jenkins

First you should have a source code management commit

You have to complete the other builds

You have to schedule it to run at a specified time

Give a manual build request.