

DEVOPS

Table of Contents

1) what is DevOps?	2
2) why do we need DevOps?	2
3) Mention the key aspects or principle behind DevOps?.....	2
4) List out some of the popular tools for DevOps?	2
5) what is a version control system?	3
6) What is Git and explain the difference between Git and SVN?.....	3
7) what language is used in Git?	3
8) what is SubGit?	3
9) how can you clone a Git repository via Jenkins?	4
10) What are the Advantages of Ansible?.....	4
11) what's the use of Ansible?	4
12) what's the difference between Ansible Playbook and Roles?.....	4
13) How do I see a list of all the ansible_ variables?	4
14) what is Docker?	5
15) what is Docker image?	5
16) what is Docker Container?.....	5
17) Can we consider DevOps as Agile methodology?.....	5
18) what are the advantages of using Git?	5
19) what is kernel?	5
20) what is difference between grep -i and grep -v?	6
21) How can you define particular space to the file.	6
22) what is concept of sudo in linux?	6
23) what is a Jenkins Pipeline?	6
24)How to stop and restart the Docker container?.....	7
25)What platforms does Docker run on?	7
26) what are the tools used for docker networking?	7
27) what is docker compose?	7
28) What is Scrum?.....	7
29) What does the commit object contain?.....	8
30) Explain the difference between git pull and git fetch?	8
31) How do we know in Git if a branch has already been merged into master?	8
32) What is 'Staging Area' or 'Index' in GIT?	8
33) What is Git Stash?	9
34) What is Git stash drop?	9
35) What is the function of 'git config'?.....	9
36) How can you create a repository in Git?	10

37) Describe the branching strategies you have used?	10
38) What is Jenkins?.....	10
39) What is the difference between Maven, Ant and Jenkins?	10
40) Explain what is continuous integration?	11
41) What is the relation between Hudson and Jenkins?	11
42) What are the advantages of Jenkins?	11
43) Which SCM tools does Jenkins supports?	11
44) How can you setup Jenkins jobs?.....	12

1) 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 an implementation of Agile methodology on Development side as well as Operations side.

2) 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.

3) 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

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

Git - Jenkins - Ansible - Puppet - Nagios - Docker - ELK (Elasticsearch, Logstash, Kibana)

5) 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 work 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

6) 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

1. Git is a Decentralized Version Control Tool SVN is a Centralized Version Control Tool
2. 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
3. Push and pull operations are fast Push and pull operations are slower compared to Git
4. It belongs to 3rd generation Version Control Tool It belongs to 2nd generation Version Control tools
5. Client nodes can share the entire repositories on their local system Version history is stored on server-side repository
6. Commits can be done offline too Commits can be done only online
7. Work are shared automatically by commit Nothing is shared automatically

7) 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.

8) 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.

9) 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.

10) What are the Advantages of Ansible?

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.

Advantage of Ansible Agentless, it doesn't require any extra package/daemons to be installed . Very low overhead . Good performance . Idempotent (replicate the same actions on all nodes) . Very Easy to learn () . Declarative not procedural

11) 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.

12) 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.	Maps among hosts and roles.
Example: common, webserver.	Example: site.yml, fooservers.yml, webserver.yml.

13) 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.

14) 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.

15) what is Docker image?

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

16) what is Docker Container?

Docker Container is the running instance of Docker Image.

17) 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.

18) 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.

19) what is kernel?

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

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

I ignore alphabet difference V accept this value

ex)

1. `ls | grep -i docker`

```
Dockerfile
docker.tar.gz
```

2. `ls | grep -v docker`

```
Desktop
Dockerfile
Documents
Downloads
```

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

21) 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
```

22) 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.

23) 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.

24)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`

25)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

26) what are the tools used for docker networking?

For docker networking we generally use kubernetes and docker swarm.

27) 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.

28) 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

29) 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).

30) 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
```

31) 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.

32) 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>
```

33) 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.

34) 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

35) 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.
```

36) 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.

37) 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.

38) 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.

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

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

40) 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.

41) 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.

42) What are the advantages of Jenkins?

Advantage of using Jenkins

1. Bug tracking is easy at early stage in development environment.
2. Provides a very large numbers of plugin support.
3. Iterative improvement to the code, code is basically divided into small sprints.
4. Build failures are cached at integration stage.
5. For each code commit changes an automatic build report notification get generated.
6. To notify developers about build report success or failure, it can be integrated with LDAP mail server.
7. Achieves continuous integration agile development and test-driven development environment.
8. With simple steps, maven release project can also be automated.

43) Which SCM tools does Jenkins supports?

Source code management tools supported by Jenkins are below:

- AccuRev
- CVS
- Subversion
- Git
- Mercurial
- Perforce
- Clearcase
- RTC

44) 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.