Top 50 Devops GIT Interview Questions and Answers Pdf

1. What is GIT ?

Answer: 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.

2. What is a repository in GIT ?

Answer: 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.

3. How can we know if a branch is already merged into master in GIT ?

Answer: We can use following commands for this purpose:

- **git branch –merged master:** This prints the branches merged into master
- **git branch –merged lists:** This prints the branches merged into HEAD (i.e. tip of current branch)
- **git branch –no-merged:** This prints the branches that have not been merged

By default this applies only to local branches.

We can use -a flag to show both local and remote branches.

Or we can use -r flag to show only the remote branches.

4. What is the purpose of git stash drop ?

Answer: 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 command.

5. What is the HEAD in GIT ?

Answer: A HEAD is a reference to the currently checked out commit.

It is a symbolic reference to the branch that we have checked out.

At any given time, one head is selected as the ‘current head’ This head is also known as HEAD (always in uppercase).

6. What is the most popular branching strategy in GIT ?

Answer: There are many ways to do branching in GIT. One of the popular ways is to maintain two branches:

master: This branch is used for production. In this branch HEAD is always in production ready state.

develop: This branch is used for development. In this branch we store the latest code developed in project. This is work in progress code.Once the code is ready for deployment to production, it is merged into master branch from develop branch.

7. What is SubGit ?

Answer: SubGit is software tool used for migrating SVN to Git. It is very easy to use. By using this we can create a writable Git mirror of a Subversion repository.

It creates a bi-directional mirror that can be used for pushing to Git as well as committing to Subversion.

SubGit also takes care of synchronization between Git and Subversion.

8. What is the use of git instaweb ?

Answer: Git-instaweb is a script by which we can browse a git repository in a web browser. It sets up the gitweb and a web-server that makes the working repository available online.

9. What are git hooks ?

Answer: 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.

10. What are the main benefits of GIT ?

Answer: There are following main benefits of GIT:

• **Distributed System:** GIT is a Distributed Version Control System (DVCS). So you can keep your private work in version control but completely hidden from others. You can work offline as well.

• **Flexible Workflow:** GIT allows you to create your own workflow. You can use the process that is suitable for your project. You can go for centralized or master-slave or any other workflow.

• **Fast:** GIT is very fast when compared to other version control systems.

• **Data Integrity:** Since GIT uses SHA1, data is not easier to corrupt.

• **Free:** It is free for personal use. So many amateurs use it for their initial projects. It also works very well with large size project.

• **Collaboration:** GIT is very easy to use for projects in which collaboration is required. Many popular open source software across the globe use GIT.

11. What are the disadvantages of GIT ?

Answer: 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.

12. How will you start GIT for your project ?

Answer: We use git init command in an existing project directory to start version control for our project. After this we can use git add and git commit commands to add files to our GIT repository.

13. What is git clone in GIT ?

Answer: In GIT, we use git clone command to create a copy of an existing GIT repository in our local.

This is the most popular way to create a copy of the repository among developers.

It is similar to svn checkout. But in this case the working copy is a full-fledged repository.

14. How will you create a repository in GIT ?

Answer: To create a new repository in GIT, first we create a directory for the project. Then we run 'git init' command. Now, GIT creates .git directory in our project directory. This is how our new GIT repository is created.

15. What are the different ways to start work in GIT ?

Answer: We can start work in GIT in following ways:

New Project: To create a new repository we use git init command.

Existing Project: To work on an existing repository we use git clone command.

16. GIT is written in which language ?

Answer: Most of the GIT distributions are written in C language with Bourne shell. Some of the commands are written in Perl language.

17. What does 'git pull' command in GIT do internally ?

Answer: In GIT, git pull internally does a git fetch first and then does a git merge.

So pull is a combination of two commands: fetch and merge.

We use git pull command to bring our local branch up to date with its remote version.

18. What is git stash ?

Answer: In GIT, sometimes we do not want to commit our code but we do not want to lose also the unfinished code. In this case we use git stash command to record the current state of the working directory and index in a stash. This stores the unfinished work in a stash, and cleans the current branch from uncommitted changes.

Now we can work on a clean working directory.

Later we can use the stash and apply those changes back to our working directory.

At times we are in the middle of some work and do not want to lose the unfinished work, we use git stash command.

19. What is the meaning of 'stage' in GIT ?

Answer: In GIT, stage is a step before commit. To stage means that the files are ready for commit.

Let say, you are working on two features in GIT. One of the features is finished and the other is not yet ready. You want to commit and leave for home in the evening. But you can commit (sap training) since both of them

are not fully ready. In this case you can just stage the feature that is ready and commit that part. Second feature will remain as work in progress.

20. What is the purpose of git config command ?

Answer: We can set the configuration options for GIT installation by using git config command.

21. How can we see the configuration settings of GIT installation ?

Answer: We can use 'git config –list' command to print all the GIT configuration settings in GIT installation.

22. How will you write a message with commit command in GIT ?

Answer: We call following command for commit with a message: $/ > git commit –m < message >

23. What is stored inside a commit object in GIT ?

Answer: 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.

24. How many heads can you create in a GIT repository ?

Answer: There can be any number of heads in a repository. By default there is one head known as HEAD in each repository in GIT.

25. Why do we create branches in GIT ?

Answer: If we are simultaneously working on multiple tasks, projects, defects or features, we need multiple branches. In GIT we can create a separate branch for each separate purpose.

Let say we are working on a feature, we create a feature branch for that. In between we get a defect to work on then we create another branch for defect and work on it. Once the defect work is done, we merge that branch and come back to work on feature branch again.

So working on multiple tasks is the main reason for using multiple branches.

26. What are the different kinds of branches that can be created in GIT ?

Answer: We can create different kinds of branches for following purposes in GIT:

Feature branches: These are used for developing a feature.

Release branches: These are used for releasing code to production.

Hotfix branches: These are used for releasing a hotfix to production for a defect or emergency fix.

27. How will you create a new branch in GIT ?

Answer: We use following command to create a new branch in GIT: $/ > git checkout –b < branchname >

28. How will you add a new feature to the main branch ?

Answer: We do the development work on a feature branch that is created from master branch. Once the development work is ready we use git merge command to merge it into master branch.

29. What is a pull request in GIT ?

Answer: A pull request in GIT is the list of changes that have been pushed to GIT repository. Generally these changes are pushed in a feature branch or hotfix branch. After pushing these changes we create a pull request that contains the changes between master and our feature branch. This pull request is sent to reviewers for reviewing the code and then merging it into develop or release branch.

30. What is merge conflict in GIT ?

Answer: A merge conflict in GIT is the result of merging two commits. Sometimes the commit to be merged and current commit have changes in same location. In this scenario, GIT is not able to decide which change is more important. Due to this GIT reports a merge conflict. It means merge is not successful. We may have to manually check and resolve the merge conflict.

31. What is the command you can use to write a commit message ?

Answer: The command that is used to write a commit message is "git commit –a". The –a on the command line instructs git to commit the new content of all tracked files that have been modified. You can use "git add< file >" before git commit –a if new files need to be committed for the first time.

32. What are the main differences between GIT and SVN ?

Answer: 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.

33. What language is used in GIT ?

Answer: GIT is fast, and 'C' language makes this possible by reducing the overhead of runtimes associated with higher languages.

34. What is the function of 'GIT PUSH' in GIT ?

Answer: 'GIT PUSH' updates remote refs along with associated objects.

35. Why GIT better than Subversion ?

Answer: 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.

36. What is "Staging Area" or "Index" in GIT ?

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

37. What is GIT stash drop ?

Answer: When you are done with the stashed item or want to remove it from the list, run the git 'stash drop' command. It will remove the last added stash item by default, and it can also remove a specific item if you include as an argument.

38. How will you know in GIT if a branch has been already merged into master ?

Answer:

Git branch—merged lists the branches that have been merged into the current branch

Git branch—-no merged lists the branches that have not been merged

39. What is the function of git clone ?

Answer: 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.

40. What is the function of 'git config' ?

Answer: The 'git config' command is a convenient way to set configuration options for your Git installation. Behaviour of a repository, user info, preferences etc. can be defined through this command.

41. What does commit object contain ?

Answer:

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 SHAI name, a 40 character string that uniquely identifies the commit object.

42. How can you create a repository in Git ?

Answer: 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.

43. What is 'head' in git and how many heads can be created in a repository ?

Answer: A 'head' is simply a reference to a commit object. In every repository, there is a default head referred as "Master". A repository can contain any number of heads.

44. What is the purpose of branching in GIT ?

Answer: 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.

45. What is the common branching pattern in GIT ?

Answer: 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.

46. What is a 'conflict' in git ?

Answer: 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.

47. How can conflict in git resolved ?

Answer: 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 merger, so it sets the parents of the commit correctly.

48. To delete a branch what is the command that is used ?

Answer: Once your development branch is merged into the main branch, you don't need development branch. To delete a branch use, the command "git branch –d [head]"

49. What is another option for merging in git ?

Answer: "Rebasing" is an alternative to merging in git.

50. What is the syntax for "Rebasing" in Git ?

Answer: The syntax used for rebase is "git rebase [new-commit]" Company