

Top 100+ Git Interview Questions and Answers [2023 Edition]

Lesson 11 of 11By Sana Afreen

Last updated on Jan 13, 2023353148

Top 100+ Git Interview Questions and Answers [2023 Edition]

[Previous](#)

[Table of Contents](#)

[What Is Git and Why Is It Used?](#)[Basic Git Interview Questions](#)[Intermediate Git Interview Questions](#)[Popular Git Interview Questions](#)[Advanced Git Interview Questions](#)[View More](#)

Git is the most popular source code management tool. Whether you're a programmer or a non-technical person, Git will help you with its collaboration features like bug tracking, task management, and wikis.

Git plays a significant role in an organization to implement DevOps methodologies. Looking at the popularity of Git in DevOps, it's quite evident that there will be many job opportunities in the near future. According to a report from Grand View Research, the global DevOps market is expected to reach 12.85 Bn by 2025, growing at a CAGR of 18.60%. Here are some of the top Git interview questions that will help you crack an interview.

Prepare to Answer all the Questions!

Caltech Program in DevOpsEXPLORE PROGRAMPrepare to Answer all the Questions!

[What Is Git and Why Is It Used?](#)

Git is an open-distributed version control system that allows developers to track changes to their codebase and collaborate on projects with other developers. Git is typically used for software development, but it can be used for any type of file.

When a developer makes a change to a code file, they can commit that change to their local Git repository. Then, they can push their changes to a remote Git repository, such as a server hosted by a company or an open-source project. Other developers can then pull the changes down from the remote repository and incorporate them into their code.

And an important thing about Git is that it is a free source and open to anyone.

Basic Git Interview Questions

1. What is Git?

Git is a version control system for tracking changes in computer files and is used to help coordinate work among several people on a project while tracking progress over time. In other words, it's a tool that facilitates source code management in software development.

Git favors both programmers and non-technical users by keeping track of their project files. It enables multiple users to work together and handles large projects efficiently.

git-1

2. What do you understand by the term 'Version Control System'?

A version control system (VCS) records all the changes made to a file or set of data, so a specific version may be called later if needed.

This helps ensure that all team members are working on the latest version of the file

server

3. What is GitHub?

To provide Internet hosting for version control and software development, GitHub makes use of Git.

4. Mention some popular Git hosting services.

GitHub

SourceForge

GitLab

Bitbucket

5. Different types of version control systems

Local version control systems have a database that stores all file changes under revision control on disc in a special format.

Centralized version control systems have a single repository, from which each user receives their working copy.

Distributed version control systems contain multiple repositories, and different users can access each one with their working copy.

6. What benefits come with using GIT?

Data replication and redundancy are both possible.

It is a service with high availability.

There can only be one Git directory per repository.

Excellent network and disc performance are achieved.

On any project, collaboration is very simple.

7. What's the difference between Git and GitHub?

Git

GitHub

Git is a software

GitHub is a service

Git can be installed locally on the system

GitHub is hosted on the web

Provides a desktop interface called git GUI

Provides a desktop interface called GitHub Desktop.

It does not support user management features

Provides built-in user management

Get Certified in DevOps with Caltech CTME

Free Webinar | Watch the WebcastEXPLORE NOW!Get Certified in DevOps with Caltech CTME

8. What is a Git repository?

Git repository refers to a place where all the Git files are stored. These files can either be stored on the local repository or on the remote repository.

git-repo

9. How can you initialize a repository in Git?

If you want to initialize an empty repository to a directory in Git, you need to enter the git init command. After this command, a hidden .git folder will appear.

taha-git

10. How is Git different from Subversion (SVN)?

GIT

SVN

Git is a distributed decentralized version control system

SVN is a centralized version control system.

Git stores content in the form of metadata.

SVN stored data in the form of files.

The master contains the latest stable release.

In SVN, the trunk directory has the latest stable release

The contents of Git are hashed using the SHA-1 hash algorithm.

SVN doesn't support hashed contents.

11. Name a few Git commands with their function.

Git config - Configure the username and email address

Git add - Add one or more files to the staging area

Git diff - View the changes made to the file

Git init - Initialize an empty Git repository

Git commit - Commit changes to head but not to the remote repository

12. What are the advantages of using Git?

Faster release cycles

Easy team collaboration

Widespread acceptance

Maintains the integrity of source code

Pull requests

git-7.

13. What language is used in Git?

Git is a fast and reliable version control system, and the language that makes this possible is 'C.'

Using C language reduces the overhead of run times, which are common in high-level languages.

Get All Your Career Growth Questions Answered!

DevOps Engineer Masters ProgramEXPLORE PROGRAMGet All Your Career Growth Questions Answered!

14. What is the correct syntax to add a message to a commit?

```
git commit -m "x files created"
```

15. Which command is used to create an empty Git repository?

git init - This command helps to create an empty repository while working on a project.

16. What does git pull origin master do?

The git pull origin master fetches all the changes from the master branch onto the origin and integrates them into the local branch.

```
git pull = git fetch + git merge origin/ master
```

After having gone through the beginner level Git interview questions, let us now look at intermediate GIT interview questions and answers.

Intermediate Git Interview Questions

17. What does the git push command do?

The Git push command is used to push the content in a local repository to a remote repository. After a local repository has been modified, a push is executed to share the modifications with remote team members.

Git-push-command

18. Difference between git fetch and git pull.

Git Fetch

Git Pull

The Git fetch command only downloads new data from a remote repository.

Git pull updates the current HEAD branch with the latest changes from the remote server.

It does not integrate any of these new data into your working files.

Downloads new data and integrate it with the current working files.

Command - git fetch origin

git fetch --all

Tries to merge remote changes with your local ones.

Command - `git pull origin master`

19. GitHub, GitLab and Bitbucket are examples of git repository _____ function?

hosting. All the three are services for hosting Git repositories

20. What do you understand about the Git merge conflict?

A Git merge conflict is an event that occurs when Git is unable to resolve the differences in code between the two commits automatically.

Git is capable of automatically merging the changes only if the commits are on different lines or branches.

merge-conflict.

Free Course: Introduction to DevOps Tools

Master the Fundamentals of DevOps Tools [ENROLL NOW](#) Free Course: Introduction to DevOps Tools

21. How do you resolve conflicts in Git?

Here are the steps that will help you resolve conflicts in Git:

Identify the files responsible for the conflicts.

Implement the desired changes to the files

Add the files using the `git add` command.

The last step is to commit the changes in the file with the help of the `git commit` command.

22. What is the functionality of git ls-tree?

The git ls-tree command is used to list the contents of a tree object.

23. What is the process to revert a commit that has already been pushed and made public?

There are two processes through which you can revert a commit:

1. Remove or fix the bad file in a new commit and push it to the remote repository. Then commit it to the remote repository using:

```
git commit -m "commit message"
```

2. Create a new commit to undo all the changes that were made in the bad commit. Use the following command:

```
git revert <commit id>
```

24. How is a bare repository different from the standard way of initializing a Git repository?

Standard way

Bare way

You create a working directory with the git init command.

Does not contain any working or checked out copy of source files.

A .git subfolder is created with all the git-related change history.

Bare repositories store git revision history in the root folder of your repository instead of the .git subfolder.

25. What does git clone do?

Git clone allows you to create a local copy of the remote GitHub repository. Once you clone a repo, you can make edits locally in your system rather than directly in the source files of the remote repo

26. What is Git stash?

Let's say you're a developer and you want to switch branches to work on something else. The issue is you don't want to make commits in uncompleted work, so you just want to get back to this point later. The solution here is the Git stash.

Git stash takes your modified tracked files and saves it on a stack of unfinished changes that you can reapply at any time. To go back to the work you can use the stash pop.

27. What does the git reset --mixed and git merge --abort commands do?

git reset --mixed is used to undo changes made in the working directory and staging area.

git merge --abort helps stop the merge process and return back to the state before the merging began.

28. What do you understand about the Staging area in Git?

The Staging Area in Git is when it starts to track and save the changes that occur in files. These saved changes reflect in the .git directory. Staging is an intermediate area that helps to format and review commits before their completion.

DevOps Engineer Master's Program

Bridge between software developers and operationsEXPLORE COURSEDevOps Engineer Master's Program

29. What is Git Bisect and how do you use it?

The Git Bisect command performs a binary search to detect the commit which introduced a bug or regression in the project's history.

Syntax: `git bisect <subcommand> <options>`

30. How do you find a list of files that has been changed in a particular commit?

The command to get a list of files that has been changed in a particular commit is:

```
git diff-tree -r {commit hash}
```

-r flag allows the command to list individual files

commit hash lists all the files that were changed or added in the commit.

31. What is the use of the git config command?

The git config command is used to set git configuration values on a global or local level. It alters the configuration options in your git installation. It is generally used to set your Git email, editor, and any aliases you want to use with the git command.

32. What is the functionality of git clean command?

The git clean command removes the untracked files from the working directory.

33. What is SubGit and why is it used?

SubGit is a tool that is used to migrate SVN to Git. It transforms the SVN repositories to Git and allows you to work on both systems concurrently. It auto-syncs the SVN with Git.

34. If you recover a deleted branch, what work is restored?

The files that were stashed and saved in the stashed index can be recovered. The files that were

untracked will be lost. Hence, it's always a good idea to stage and commit your work or stash them.

35. Explain these commands one by one— git status, git log, git diff, git revert <commit>, git reset <file>.

Git status - It shows the current status of the working directory and the staging area.

Git revert<commit> - It is used for undoing changes to a repository's commit history.

Git log- It is a key tool for reviewing and reading the history of everything that happens to a repository.

Git diff- It is a multi-purpose Git command that performs a diff function on Git data sources when executed.

Git reset<file>- it is used to unstage a file.

36. What exactly is tagging in Git?

Tagging enables developers to mark all significant checkpoints as their projects progress.

37. What exactly is forking in Git?

It is a repository duplicate and forking allows one to experiment with changes without being concerned about the original project.

38. How to change any older commit messages?

You can change the most recent commit message with the git commit —amend command.

39. How to handle huge binary files in Git?

Git LFS is a Git extension for dealing with large and binary files in a separate Git repository.

40. Name a few GIT tools.

Git comes with a few built-in tools like Git Bash and Git GUI.

41. Will you make a new commit or amend an existing one?

The git commit —amend command allows you to easily modify the most recent commit.

42. What do you mean by branching strategy?

It is employed by a software development team while writing and managing code with a version control system.

43. Difference between head, working tree, and index.

They are all names for various branches. Even Though a single git repository can track an arbitrary number of branches, the working tree is only associated with one of them, and HEAD points to that branch.

44. Is there a git GUI client available for Linux?

Git includes built-in GUI tools for committing (git-gui) and browsing (gitk), but there are a number of third-party tools available for users seeking platform-specific experience.

45. What is the benefit of a version control system?

Version control enables software teams to maintain efficiency and agility while the team grows by adding more developers

DevOps Engineer Exam Practice Test

Assess your understanding of the DevOps conceptsTRY ANSWERINGDevOps Engineer Exam Practice Test

46. What do you mean by git instaweb?

It is a script used to set up a temporary instance of Gitweb.

47. What exactly is the forking workflow?

Forking is a git clone operation that is performed on a server copy of a project's repository.

48. Mention benefits of forking workflow.

Contributions can be integrated without everyone trying to push to a single central repository.

49. What is the Gitflow workflow?

The Gitflow Workflow specifies a strict branching model centered on the project release.

50. What does the commit object contain?

The commit object contains a tree of blob objects and other tree objects that represent the project revision's directory structure.

51. Write the syntax of rebasing in git.

Syntax is as follows: `$git rebase <branch name>`

52. What are Git Hooks?

They are scripts that are executed automatically whenever a specific event occurs in a Git repository.

53. What is Git stash vs Git stash pop?

Git stash pop removes the (topmost, by default) stash when applied, whereas git stash apply keeps it in the stash list for future use.

54. Explain git reflog This command is used by Git to record changes made to the branches' tips.

55. Role of the git annotate command.

In git, it is used to track each line of the file based on the commit information.

56. What is a git Directory?

It is the storage place of the metadata and object database of the project.

57. How can a conflict be settled in Git?

Edit the files to resolve any incompatible changes first, then use "git add" to add the corrected files and "git commit" to save the repaired merge.

58. What is the standard method for branching in GIT?

In GIT, the best way to create a branch is to have one 'main' branch and then another branch for implementing the changes that we want to make.

FREE DevOps Certification Training

Master the fundamentals of DevOps [START LEARNING](#) FREE DevOps Certification Training

59. How do you set up a Git repository?

If you want to add an empty repository to a directory in Git, use the git init command.

60. What is the proper syntax for appending a message to a commit?

Git commit -m "x files created" is the syntax.

61. Use of git instaweb.

It is used to launch a web browser and a webserver with an interface into a local repository automatically.

62. Describe git ls-tree.

It represents a tree object with each item's mode and name included.

63. What exactly is git cherry-pick?

A command typically used to move specific commits from one branch of a repository to another.

64. State the difference between “git remote” and “git clone”?

“Git remote” allows you to create an entry in the git configuration which specifies a URL.

“Git clone” lets you create a new git repository by letting you copy it from the current URL.

65. Difference between “pull request” and “branch”?

“Pull request” is done when you feel like changing the developer’s change to another person’s code branch. And “Branch” is just a separate version of code.

66. How might you recover a branch that has previously pushed changes in the main repository yet has been coincidentally erased from each team member’s local machines?

We can easily recover this by seeing the latest commit of the branch in the reflog and then going through the new branch.

67. What is a detached head?

Detached head refers to that the currently checked repository is not in the local branch.

68. What command helps us to know the branches merged into master and which are not?

`git branch --merged` lets us get the list of the branches which are currently merged into the current branch

`git branch --no-merged` shows the branches which are not merged

69. Is LDAP Authentication Supported?

GitLab API only supports LDAP authentication since version 6.0 and higher.

Now let’s raise the level of difficulty with advanced Git interview questions and answers.

Interested to learn more about Git? Check out the DevOps Engineer Master's Program and get certified today.

Popular Git Interview Questions

70. A simple definition of Git

Git is a free and open-source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

71. Is the C++ language used in Git?

Yes, but C programming language is a widely used language in Git.

72. How does Git work?

Git works by tracking changes to files in a project and allowing developers to easily revert to previous versions if necessary. Git also makes it easy to collaborate on projects, as it allows multiple developers to work on the same codebase simultaneously.

73. What are some of the most popular Git commands?

Some of the most popular Git commands are,

"git init" (which initializes a Git repository)

"git add" (which adds files to a Git repository)

And "git commit" (which saves changes to a Git repository).

74. List out the functions provided by Git repository hosting service

There are many functions that a Git repository hosting service can provide. Some of the most common and useful functions include:

Providing a web interface for users to interact with the repositories

Allowing users to clone repositories

Allowing users to view and download statistics about the repositories

Providing a way for users to push changes to the repositories

Keeping track of changes made to the repositories

75. What is the difference between Git and other revision control systems?

Git is a distributed revision control system, which means that it can be used without a central server. This allows for a great deal of flexibility in how projects are managed.

On the other side, revision control systems are often centralized, which can limit the flexibility of how projects are managed.

76. How does Git work?

Git works by tracking changes to files in a repository. When a file is changed, Git calculates a unique identifier for the change, called a "commit hash". The commit hash allows Git to identify the change and track it over time.

77. How do I install Git?

Installing Git is simple. Just download the latest version from the Git website (<https://git-scm.com/>).

78. How do I use Git?

To use Git, a developer first creates a local repository on their computer. This repository contains all the files for a project and the history of all the changes made to those files.

Or just follow the instructions in the Git documentation (<https://git-scm.com/doc>).

79. What are some of the drawbacks of Git?

One of the main drawbacks is that it can be difficult to learn and use, especially for those who are not familiar with version control systems.

And Git is not always reliable and can sometimes be slow.

FREE GIT Training

Learn the basics of GITENROLL NOWFREE GIT Training

80. What are some of the most important commands in Git?

Some of the most important Git commands are "commit", "push", and "pull".

The "commit" command is used to save changes to the local repository.

And the "push" command is used to send changes to the remote repository.

Then the "pull" command is used to retrieve changes from the remote repository.

81. What are some of the most important features of Git?

Some of the most important features of Git are its distributed nature, its ability to track changes, and its support for branches.

The distributed nature of Git allows developers to work independently and offline.

The ability to track changes helps developers to keep track of their work and revert to previous versions if necessary.

The support for branches allows developers to experiment with new features without affecting the main codebase.

82. What is a branch in Git?

A branch is a way to isolate development work on a particular aspect of a project. When a branch is created, it diverges from the primary branch. It allows developers to work on a new feature or bug fix without affecting the main codebase.

83. What is a commit in Git?

A commit is a way to save changes to a branch. When a commit is made, a snapshot of the current state of the branch is created. This snapshot can be used to revert the branch to that state if necessary.

84. What is conflict in Git?

Conflict in Git occurs when two or more developers have made changes to the same part of a file, and those changes can't be automatically merged. When this happens, Git will mark the file as conflicted and leave it up to the developers to resolve the conflict.

Resolving a conflict can be done by manually editing the file to choose which changes should be kept, or by using a tool like Git's merge command to automatically merge the changes.

85. What does the git status command do?

The git status command is used to obtain the current state of a Git repository. This command can be used to determine whether the repository is clean or dirty, and to see which files have been modified. The git status command will also show which branch is currently checked out and whether there are any uncommitted changes.

86. Why is it considered to be easy to work on Git?

There are many reasons that Git is considered an easy tool to work with.

It has a straightforward learning curve. Even those new to programming can easily learn how to use Git with just a few hours of practice.

Git is highly flexible and can be easily customized to fit the needs of any project.

And, Git is very stable and reliable, so users can trust that their work will be safe and sound.

87. What do you know about Git Stash?

Git stash is a powerful tool that allows you to save your changes and revert your working directory to a previous state. This is especially useful when switching branches or reverting to a previous commit.

And Git Stash takes a snapshot of your changes and stores them away for later use.

88. What differentiates between the commands git remote and git clone?

The main difference between the git remote and git clone commands is that the git remote adds a remote repository as a shortcut to your current repository, while the git clone creates an entirely new copy of a remote repository.

89. Tell me the difference between git pull and git fetch?

Both of these commands will fetch any new commits from the remote repository, but they differ in how they handle these commits.

Git pull will merge the remote commits into the current branch, while git fetch will simply retrieve the commits and store them in the local repository. This means that if you have any uncommitted changes, git pull may result in merge conflicts, while git fetch will not.

90. Is Git and GitHub the same thing?

No, Git and GitHub are two different things.

Git is a version control system that lets you track changes to your code.

GitHub is a hosting service for Git repositories. You can use GitHub to store your code remotely, or you can use it to collaborate with other developers on a project.

91. What about Git reflog?

Git reflog is a history of all the changes made to a git repository. It is a valuable tool for debugging and troubleshooting purposes.

And Git reflog can be used to view the history of a repository, see who made what changes, and when those changes were made.

92. What is a detached head?

A detached HEAD is a state where the HEAD pointer is not pointing to the current commit. This can happen if you check out a commit that is not the most recent, or if you reset your head to a previous commit.

93. How to avoid a detached head?

There are a few different ways to avoid a detached HEAD.

The first is to simply commit your changes before switching branches. This will ensure that your changes

are saved to a specific branch, and you won't have to worry about them being lost when you switch branches.

Another way to avoid detached HEAD is to use the "git checkout" command with the "-b" option.

94. How will you resolve conflict in Git?

To resolve a conflict in Git, you will need to first identify the source of the conflict. Once you have identified the source of the conflict, you can use the "git pull" command. This will pull the latest changes from the remote repository and merge them with your local copy.

If the "git pull" command doesn't resolve the conflict, you can try the "git merge" command. This will merge the two versions of the code manually. You will need to resolve the conflicts manually and then commit the merged code.

95. What is Subgit and where do you use Subgit?

Subgit is a tool for managing Git repositories with Subversion history. It allows you to keep your existing subversion history while moving to Git, and it also provides a way to keep your Git history synchronized with subversion.

There are many reasons that you might want to use Subgit,

For example, if you have a large subversion repository with a lot of history, moving to Git can be a huge undertaking; that's where Subgit can help transition smoother.

And if you work in an environment where subversion is the primary version control system, using Subgit can help you keep your Git history in sync with the rest of the team.

Advanced Git Interview Questions

96. Explain the different points when a merge can enter a conflicted stage.

There are two stages when a merge can enter a conflicted stage.

1. Starting the merge process

If there are changes in the working directory of the stage area in the current project, the merge will fail to start. In this case, conflicts happen due to pending changes that need to be stabilized using different Git commands.

2. During the merge process

The failure during the merge process indicates that there's a conflict between the local branch and the branch being merged. In this case, Git resolves as much as possible, but some things have to be fixed manually in the conflicted files.

97. What has to be run to squash the last N commits into a single commit?

In Git, squashing commits means combining two or more commits into one.

Use the below command to write a new commit message from the beginning.

```
git reset -soft HEAD~N &&git commit
```

But, if you want to edit a new commit message and add the existing commit messages, then you must extract the messages and pass them to Git commit.

The below command will help you achieve this:

```
git reset -soft HEAD~N &&git commit -edit -m"${git log -format=%B -reverse .HEAD@{N}}"
```

98. What is the difference between fork, branch, and clone?

Fork

Branch

Clone

The fork is the process when a copy of the repository is made. It's usually experimentation in the project without affecting the original project. They're used to advise changes or take inspiration from someone else's project.

Git branches refer to individual projects within a git repository. If there are several branches in a repository, then each branch can have entirely different files and folders.

Git clone refers to creating a clone or a copy of an existing git repository in a new directory. Cloning automatically creates a connection that points back to the original repository, which makes it very easy to interact with the central repository.

99. How is Git merge different from Git rebase?

Git merge is used to incorporate new commits into your feature branch.

feature-master

Git merge creates an extra merge commit every time you need to incorporate changes.

It pollutes your feature branch history.

As an alternative to merging, you can rebase the feature branch into master.

feature-master2

Git rebase Incorporates all the new commits in the master branch.

It rewrites the project history by creating brand new commits for each commit in the original branch

100. What is the command used to fix a broken commit?

To fix a broken commit in Git, you may use the “git commit --amend” command, which helps you combine the staged changes with the previous commits instead of creating an entirely new commit.

101. How do you recover a deleted branch that was not merged?

To recover a deleted branch, first, you can use the git reflog command. It will list the local recorded logs for all the references. Then, you can identify the history stamp and recover it using the git checkout command.