



DevOps Git Interview Questions

Basics of Git:

1. What is Git?

- **Answer:** Git is a distributed version control system that allows multiple developers to collaborate on a project, tracking changes to source code during software development.

2. Explain the difference between Git and SVN.

- **Answer:** Git is a distributed version control system, while SVN (Subversion) is a centralized version control system. In Git, each developer has a local copy of the entire repository, enabling offline work and faster commits.

3. What is a repository in Git?

- **Answer:** A repository in Git is a collection of files and versioning information. It contains the entire history of the project and is typically hosted on a remote server.

4. How do you initialize a new Git repository?

- **Answer:** Use the command `git init` to initialize a new Git repository.

5. What is the purpose of the .gitignore file?

- **Answer:** The .gitignore file specifies files and directories that should be ignored by Git. These are typically files generated during the build process or containing sensitive information.

Working with Commits:

6. Explain what a commit is in Git.

- **Answer:** A commit in Git represents a snapshot of changes made to a repository. It includes a unique identifier (SHA-1 hash), author information, timestamp, and a reference to the parent commit.

7. How do you create a new commit?

- **Answer:** Use the command `git commit -m "Commit message"` to create a new commit with a descriptive message.

8. What is the purpose of the staging area (index) in Git?

- **Answer:** The staging area is an intermediate step where changes are marked for inclusion in the next commit. It allows selective commit of specific changes.

9. Explain the difference between `git pull` and `git fetch`.

- **Answer:** `git pull` fetches changes from a remote repository and merges them into the current branch. `git fetch` only retrieves changes but does not automatically merge them.

10. How do you undo the last commit in Git?

- **Answer:** Use the command `git reset HEAD^` to undo the last commit. This command keeps changes in your working directory.

Branching and Merging:

11. What is a branch in Git?

- **Answer:** A branch in Git is a lightweight movable pointer to a commit. It allows for the creation of separate lines of development.

12. How do you create a new branch?

- **Answer:** Use the command `git branch branch_name` to create a new branch.

13. Explain the difference between `git merge` and `git rebase`.

- **Answer:** `git merge` combines changes from different branches, creating a new commit. `git rebase` moves or combines a sequence of commits to a new base commit.

14. What is a merge conflict? How do you resolve it?

- **Answer:** A merge conflict occurs when Git cannot automatically merge changes. To resolve it, manually edit the conflicting files, mark them as resolved with `git add`, and then complete the merge with `git merge --continue`.

15. What is the purpose of Git tags?

- **Answer:** Git tags are references to specific points in Git history, typically used to label release versions. They provide a stable reference to a specific commit.

Remote Repositories:

16. How do you clone a Git repository?

- **Answer:** Use the command `git clone repository_url` to clone a Git repository from a remote server.

17. Explain the difference between `git push` and `git pull`.

- **Answer:** `git push` sends local changes to a remote repository. `git pull` retrieves changes from a remote repository and merges them into the local branch.

18. What is the purpose of `git remote`?

- **Answer:** `git remote` is used to manage connections to remote repositories. It lists, adds, or removes remote repositories.

19. How do you change the remote repository URL in Git?

- **Answer:** Use the command `git remote set-url origin new_url` to change the remote repository URL.

20. What is a Git fork, and how is it different from a clone?

- **Answer:** A fork is a copy of a repository on a user's GitHub account. It's often used to propose changes to someone else's project. A clone is a copy of a repository on the local machine.

Git Workflow:

21. Explain the Gitflow workflow.

- **Answer:** Gitflow is a branching model that defines a standard repository structure and branching strategy, including development, feature, release, and hotfix branches.

22. What is a pull request?

- **Answer:** A pull request is a way to propose changes to a repository. It allows collaborators to review and discuss the changes before merging them.

23. How do you squash commits in Git?

- **Answer:** Use the interactive rebase with the command `git rebase -i HEAD~n`, where `n` is the number of commits. Squash or fixup commits during the rebase.

24. What is the purpose of Git hooks?

- **Answer:** Git hooks are scripts that run automatically at key points in the Git workflow. They allow customizing and automating actions such as pre-commit checks or post-receive deployments.

25. Explain the difference between a fast-forward and a non-fast-forward merge.

- **Answer:** A fast-forward merge occurs when the branch being merged has no new commits since the last commit in the base branch. A non-fast-forward merge creates a new commit, even if the branch being merged has no new commits.

Advanced Git:

26. What is `git bisect` used for?

- **Answer:** `git bisect` is used to find the commit that introduced a bug. It performs a binary search through the commit history to locate the faulty commit.

27. How do you change the last commit message in Git?

- **Answer:** Use the command `git commit --amend -m "New commit message"` to change the last commit message.

28. Explain the difference between `git cherry-pick` and `git rebase`.

- **Answer:** `git cherry-pick` applies specific commits to the current branch. `git rebase` moves or combines a sequence of commits.

29. What is Git LFS (Large File Storage)?

- **Answer:** Git LFS is an extension to Git that handles large files by replacing them with text pointers, storing the actual file content in an external storage system.

30. How do you remove a file from Git without deleting it locally?

- **Answer:** Use the command `git rm --cached filename` to remove a file from Git without deleting it locally.

Git Commands:

31. What does the command `git status` do?

- **Answer:** `git status` shows the status of changes as untracked, modified, or staged. It also displays information about the branch and the presence of uncommitted changes.

32. How do you show the commit history in Git?

- **Answer:** Use the command `git log` to show the commit history. Options like `--oneline` and `--graph` provide more concise or graphical views.

33. What is the purpose of `git cherry-pick`?

- **Answer:** `git cherry-pick` is used to apply specific commits from one branch to another.

34. How do you create a new branch and switch to it in one command?

- **Answer:** Use the command `git checkout -b branch_name` to create and switch to a new branch.

35. What does the `git reflog` command do?

- **Answer:** `git reflog` shows a log of all reference updates, including branch switches and commit changes. It helps recover lost commits.

Collaboration and Code Review:

36. What is a Git submodule?

- **Answer:** A Git submodule allows including another Git repository as a subdirectory within a parent repository. It enables versioning and tracking of external dependencies.

37. How do you resolve a conflict in a pull request?

- **Answer:** To resolve a conflict in a pull request, fetch the changes locally, create a new branch, resolve conflicts, commit changes, push the branch, and update the pull request.

38. Explain the difference between a fork and a branch.

- **Answer:** A fork is a copy of a repository on a user's GitHub account, while a branch is a separate line of development within a repository.

39. What is the purpose of code review in Git?

- **Answer:** Code review in Git involves having team members review changes before merging. It helps maintain code quality, catch bugs, and ensure adherence to coding standards.

40. How do you revert a commit that has already been pushed to a remote repository?

- **Answer:** Use the command `git revert commit_sha` to create a new commit that undoes the changes introduced by the specified commit.

Git Configurations:

41. How do you set up Git globally on your machine?

- **Answer:** Use the commands:
- `git config --global user.name "Your Name"`
- `git config --global user.email your.email@example.com`

42. What is the purpose of .gitconfig file?

- **Answer:** The `.gitconfig` file stores configuration settings for a user's Git installation, including user details, aliases, and other preferences.

43. How do you configure Git to use a specific text editor?

- **Answer:** Use the command `git config --global core.editor "editor_name"` to configure Git to use a specific text editor.

44. What is the purpose of Git hooks, and where are they stored?

- **Answer:** Git hooks are scripts that run at specific points in the Git workflow. They are stored in the `.git/hooks` directory of a Git repository.

45. How do you list all configured remote repositories in Git?

- **Answer:** Use the command `git remote -v` to list all configured remote repositories along with their URLs.

Git Security:

46. Explain the concept of Git SSH keys.

- **Answer:** Git SSH keys are used for secure authentication between a Git client and a remote repository. They ensure secure communication without the need for entering a password.

47. How do you sign commits in Git?

- **Answer:** Use the command `git commit -S -m "Commit message"` to sign commits with a GPG key.

48. What is Git Credential Manager?

- **Answer:** Git Credential Manager is a tool that securely stores Git credentials, providing a convenient way to authenticate without entering passwords repeatedly.

49. How do you change the Git remote URL from HTTPS to SSH?

- **Answer:** Use the command `git remote set-url origin git@github.com:user/repo.git` to change the remote URL from HTTPS to SSH.

50. What is the purpose of Git stash, and how do you use it?

- **Answer:** Git stash is used to temporarily save changes that are not ready to be committed. Use `git stash` to save changes and `git stash apply` to retrieve them later.

Git Internals:

51. Explain the Git object model.

- **Answer:** The Git object model consists of four main object types: blobs (file content), trees (directory structure), commits (snapshot and metadata), and tags (references to commits).

52. What is the difference between a shallow clone and a deep clone in Git?

- **Answer:** A shallow clone fetches only a limited number of commit history, while a deep clone fetches the entire commit history.

53. How does Git handle branching and merging internally?

- **Answer:** Git handles branching and merging through a directed acyclic graph (DAG). Each commit points to its parent commit(s), forming a history tree.

54. What is the Git index, and how does it work?

- **Answer:** The Git index (staging area) is a file that contains information about files to be included in the next commit. It acts as a bridge between the working directory and the repository.

55. Explain the concept of Git objects.

- **Answer:** Git objects are the fundamental units in Git, representing different types of data, such as files, directories, commits, and tags. Objects are stored in the Git repository.

Git and CI/CD:

56. How does Git integrate with CI/CD pipelines?

- **Answer:** Git integrates with CI/CD pipelines by triggering builds or deployments based on events such as pushes to specific branches or the creation of pull requests.

57. What is GitOps, and how does it relate to Git?

- **Answer:** GitOps is a set of practices for managing infrastructure and application deployments using Git repositories as the single source of truth. Changes in Git trigger automated operations.

58. How do you manage secrets in a Git repository?

- **Answer:** Secrets should not be stored directly in a Git repository. Use environment variables, encrypted files, or a dedicated secrets management tool.

59. Explain the role of Git in a microservices architecture.

- **Answer:** Git is used to version control microservices codebases, enabling collaboration among development teams. It also plays a role in managing configurations and deployment scripts.

60. How do you handle versioning in a Git project?

- **Answer:** Versioning in Git is often managed using tags. Developers can create tags to mark specific commits as release versions, making it easy to reference and deploy specific points in history.