

Name: Mrunal Itankar

1. What is a distributed VCS?

- **Distributed Version Control** (also known as **distributed revision control**) is a form of version control in which the complete codebase, including its full history, is mirrored on every developer's computer.

2. What are the benefits of using Version Control System?

Traceability

Traceability is a mechanism that provides evidence of all revisions and changes made over a while. It enables users to identify the development of the file through its various stages. It tracks the contributions made by several developers.

Document History

The history of the document provides invaluable information about the author and the date of editing.

Branching And Merging

Version control allows team members to work on the same document concurrently but independent of each other without affecting the contribution of fellow collaborators.

Identity

Version control enables the current team to analyse the deletion, editing, and creation of datasets made since the original copy. It brings clarity to the development of the software.

Reduction Of Duplication and Errors

Version control can reduce the duplication of multiple and/ or out-dated versions of any given document. It, in turn, will reduce errors that surface from conflicting information displayed through multiple documents.

Management Overview

Version control enables management to have a comprehensive perspective of the development of the project.

Efficiency

Version control promotes an efficient progression of the document. Teams work to simplify complex processes and, thus, create greater scope for automation and consistency.

3. What language is used in Git?

- Git was designed as a set of programs written in **C** and several shell scripts that provide wrappers around those programs. Although most of those scripts have since been rewritten in C for speed and portability, the design remains, and it is easy to chain the components together.

4. Mention the various Git repository hosting functions.

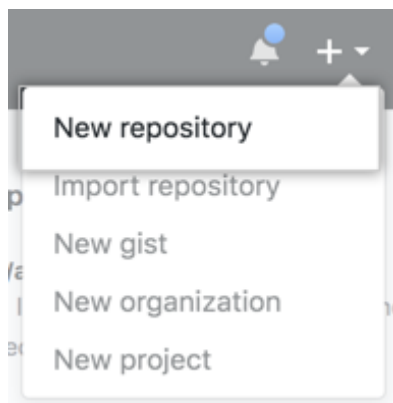
- GitHub
- GitLab
- Bitbucket
- SourceForge

5. What is a repository in Git?

- Repositories in GIT contain a collection of files of various different versions of a Project. These files are imported from the repository into the local server of the user for further updates and modifications in the content of the file. A VCS or the Version Control System is used to create these versions and store them in a specific place termed as a repository.

6. How can you create a repository in Git?

1. In the upper-right corner of any page, use the drop-down menu, and select **New repository**.

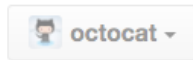


2. Type a short, memorable name for your repository. For example, "hello-world".

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner



Repository name

hello-world



Great repository names are short and memorable. Need inspiration? How about **potential-eureka**.

Description (optional)

3. Optionally, add a description of your repository. For example, "My first repository on GitHub."

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner



Repository name

hello-world



Great repository names are short and memorable. Need inspiration? How about **potential-eureka**.

Description (optional)

My first repository on GitHub

4. Choose a repository visibility. For more information, see "[About repositories](#)."

Description (optional)



Public



Anyone can see this repository. You choose who can commit.



Internal



Octo Corp **enterprise members** can see this repository. You choose who can commit.





Private



You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

5. Select **Initialize this repository with a README**.

- ☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.
- ☐  **Private**
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

- ☒ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer.

Add .gitignore: None ▾

Add a license: None ▾



Create repository

6. Click **Create repository**.

This will let you immediately clone the repository to your computer.

Add .gitignore: None ▾

Add a license: None ▾



Create repository

7. What is 'staging area' or 'index' in Git?

- The **staging area** can be described as a preview of your next commit therefore when you create a git commit, Git takes changes that are in the staging area and make them as a new commit. The staging area can be considered as a real area where git stores the changes and it also allows to add and remove changes from it.

8. What is git stash?

- Git stash saves the uncommitted changes locally, allowing you to make changes, switch branches, and perform other Git operations. You can then reapply the stashed changes when you need them. A stash is locally scoped and is not pushed to the remote by git push.

9. What is the difference between the 'git diff' and 'git status'?

- Diff command is used in git to track the difference between the changes made on a file.
- The git status command will display the whole current status of your working tree (files that are staged, modified, deleted, untracked etc.)

10. What is the difference between 'git remote' and 'git clone'?

- git remote add just creates an entry in your git config that specifies a name for a particular URL. You must have an existing git repo to use this. git clone creates a new git repository by copying an existing one located at the URI you specify.

11. What does a commit object contain?

- The commit object contains the top-level tree object hash, parent commits hash (if any), author and committer information, commit date and commit message.

12. How to remove a file from git without removing it from your file system?

- Add all the files, individually or in a folder, that you want to remove from the repo but keep locally to . gitignore.
- Execute git rm --cached put/here/your/file. ext for each file or git rm --cached folder/* if they are in a folder. ...
- Commit your changes.
- Push to remote.

13. Can you explain the Gitflow workflow?

- The Gitflow Workflow defines a strict branching model designed around the project release. This workflow doesn't add any new concepts or commands beyond what's required for the Feature Branch Workflow. Instead, it assigns very specific roles to different branches and defines how and when they should interact.

14. What are the different ways you can refer to a commit?

- In Git each commit has a unique hash. These hashes are used to identify the corresponding commits in various scenarios, for example, while trying to checkout a particular state of the code using the git checkout {hash} command.
- Along with this, Git maintains a number of aliases to certain commits, known as refs. Also, every tag that is created in the repository effectively becomes a ref and that is exactly why you can use tags instead of committing hashes in various git commands. Git also maintains a number of special aliases that are changed based on the state of the repository, such as HEAD, FETCH_HEAD, MERGE_HEAD, etc.
- In Git, commits are allowed to be referred to as relative to one another. In the case of merge commits, where the commit has two parents, ^ can be used to select one of the two parents, for example, HEAD^2 can be used to follow the second parent.

- And finally, refspecs are used to map local and remote branches together. However, these can also be used to refer to commits that reside on remote branches allowing one to control and manipulate them from a local git environment.

15. Explain the difference between reverting and resetting.

- **git revert:** It is used to remove the commits from the remote repository. Since now our changes are in the working directory, let's add those changes to the staging area and commit them.
- **git reset:** It is used when we want to unstage a file and bring our changes back to the working directory. git reset can also be used to remove commits from the local repository.