**Understanding Git and GitHub: Key Concepts and Commands**

**1) What is Git and Why is it Used?**

Git is a collaboration and distributed version control system. It stores source code, tracks file changes, and maintains project history.

**2) Git and GitHub Differences**

**Git:**

* A version control system that is installed on your local machine.

**GitHub:**

* A web-based hosting platform for Git repositories, acting as a central repository. It offers additional features like collaboration, pull requests, issue tracking, and integration with other services.

**3) What is a Repository in Git?**

A repository server has a central location where code is stored, managed, and shared with team members, enabling collaboration. Each repository represents a project and is a collection of files, directories, and their version history.

**4) Commands to Install Git and Check the Version**

$ yum install git # for RHEL

$ git --version

git version 2.43.5

**5) How to Create a Git Repository**

**i) For a New Project**

$ git init # Initializes the Git repository (.git)

$ ls -a

. .. .git # .git is the local repository

**ii) For an Existing Project**

$ git clone <repo url> # GitHub repository URL

In both cases, the .git directory will be created.

**6) Difference Between Git Commit and Git Push**

**Git Commit:** Used to save changes in the local repository, create a commit ID, and record changes. As a best practice, use a commit message summarizing the changes made.

**Git Push:** Pushes the committed changes from the local repository to the central repository. Changes are not reflected in the central repository until the git push command is used.

**7) Git Branch and Why it is Used**

Branches are lightweight, movable pointers that allow you to work on code changes without affecting the existing code. They facilitate collaboration and parallel development and help isolate work on new features.

**8) How to Create a Git Branch and Switch Between Branches**

**i) If Files Are Already Created and Committed in the Master Branch**

$ git branch <branch name>

**ii) If Files Are Not Created and Committed in the Master Branch**

$ git checkout -b test2 # Creates a new branch and switches to it

Switched to a new branch 'test2'

$ git branch

master

test1

\* test2

To switch between branches:

i)$ git checkout test1

Switched to branch 'test1'

$ git branch

master

\* test1

test2

ii)$ git switch test1

Switched to branch 'test1'

$ git branch

master

\* test1

test2

**9) How to Rename and Delete a Git Branch**

**Rename Branch:**

**i) Checkout to the branch you want to rename:**

$ git branch

master

\* test1

test2

$ git branch -m firstbranch

$ git branch

\* firstbranch

master

test2

**ii) To run from any branch:**

$ git branch -m <oldname> <newbranch name>

**Delete Branch Locally:**

$ git branch -d <branch> # Deletes branch only if it has been pushed and merged with the remote branch

$ git branch -D <branch> # Deletes branch even if it hasn't been pushed or merged yet

**Delete Branch Remotely:**

$ git push origin --delete <branchname>

**10) How to Add a Remote Repository to Local Repository**

$ git remote add origin <https remote url> # 'origin' is the alias for our remote repo

$ git remote -v # Lists the remote repositories

origin https://github.com/myselfsoundarya/intro.git (fetch)

origin https://github.com/myselfsoundarya/intro.git (push)

**11) How to Change the Locally Added Remote Repo URL**

$ git remote set-url origin <https url>

$ git remote set-url origin https://<access token>@github.com/username/directory # Command to pass the access token in the HTTPS URL

**12) How to Use SSH Keys Based Repository**

1. Generate the SSH keys on the server.
2. Add the key in GitHub:
   * Top right corner, click your account image and select **Settings**.
   * Go to **SSH and GPG Keys** and add the public key.
3. Test the connection:

$ ssh -T git@github.com

Hi myselfsoundarya! You've successfully authenticated, but GitHub does not provide shell access.

**13) What is Git Merge**

Git merge is used to combine changes from two branches into the current branch.

$ git merge <branch name> # Checkout to the branch where code needs to be merged