SRE Training (Day 5) - GIT Overview

Version Control System (VCS): A tool designed to track changes to files over time, facilitating collaboration among multiple individuals. It keeps a record of modifications and allows easy rollback to previous versions when necessary.

GIT: A distributed version control tool, while **GitHub** is a cloud platform that integrates with GIT.

I. GIT CONFIGURATION

- **Setting up User Information Locally**:
- git config user.name "Bhoomi kaushik"
- git config user.email "kaushikbhoomi007@gmail.com"

This applies only to the current local repository.

- **Setting up Global User Information**:
- git config --global user.name "Bhoomi Kaushik"

This applies to all repositories on your system.

- **Listing Configuration Details**:
- git config --list

II. STARTING YOUR FIRST REPOSITORY

1. **Creating a GitHub Repository**: To create a new repository on GitHub, follow these steps:
   - Go to GitHub and create a new repository.
   - Choose a name for your repository.
   - Optionally, add a README file that contains details about the repository.
2. **Cloning the Repository to Your Local Machine**:
   - Copy the HTTPS link for your repository.
   - Use the command:
   - git clone https://github.com/bhoomikaushik/Practice-1.git

This copies the remote repository to your local system.

3. **Creating, Adding, Committing, and Pushing a New File**:
   - To create a new file, you can use the touch command or echo with redirection, among other methods.
   - To track the changes made to this file, use:

   - git add <filename>
   - Then, commit the changes with a message:
   - git commit -m "Your commit message"
   - Finally, push the changes to the remote repository:
   - git push -u origin main

4. **Working with Branches and Pushing Changes**:
   - To rename the default master branch to main, use:
   - git branch -M main
   - After committing your changes, use the following to push them to the remote repository:
   - git push -u origin main
   - If the remote repository was not previously linked, add the remote repository URL:
   - git remote add origin <repo_url>

5. **Fetching and Pulling Changes**:
   - To retrieve changes from the remote repository:
     - git fetch fetches the changes but does not merge them.
     - git pull fetches the changes and merges them into your local branch.

Difference between git fetch and git pull:

   - git fetch: Downloads changes but does not apply them to the working directory.
   - git pull: Downloads and applies the changes, effectively updating your local repository.
   - **Example of git fetch**: A new file, file-2.txt, was added to the remote repository. After running git fetch, the commit ID changes (e.g., from b0f991a to 2bcb642), but the file doesn't appear locally.
   - **Example of git pull**: After executing git pull, the changes are fetched and merged into the local repository, and the new file (file-2.txt) appears in the working directory.

III. BRANCHES IN GIT

A branch in Git is a separate version of your project, allowing you to work on features or fixes without impacting the main codebase.

- **List branches**:
   - git branch: Lists local branches.

   - git branch -a: Lists all local and remote branches.
   - git branch -r: Lists only remote branches.

1. **Creating a New Branch**: To create a new branch:
2. git branch <branch_name>
3. **Merging Changes**: Merging can be done via a pull request on GitHub. Alternatively, to merge branches locally:
   - Switch to the branch you want to merge into (usually main).
   - Run:
   - git merge <branch_name>
   - Push the merged changes to the remote repository:
   - git push

IV. GIT STASH

git stash temporarily saves uncommitted changes, allowing you to switch branches or perform other actions without losing your progress.

- **To apply stashed changes**:
- git stash pop

This restores the latest stashed changes and removes them from the stash.

KEY LEARNINGS

1. **git push vs git push -u origin main**:
   - Both commands push local changes to the remote repository.
   - The key difference is that git push -u origin main is used for the first push to set the upstream reference, after which git push can be used for subsequent pushes without specifying origin main.
2. **git reset vs git revert**:
   - git reset: Discards all local changes and resets the working directory to a specific commit. For example:
   - git reset <commit_id>

This will change the HEAD reference and may require a git push --force to reflect the changes on the remote repository.

   - **git revert** is a safer option because it creates a new commit that undoes previous changes, maintaining the history. This is preferable to resetting, as it doesn’t alter the commit history.

This summary encapsulates the major points covered in the training session on GIT and version control basics.