

Experiment No. 1:

To perform Installation, Initialization, Configuration, and First Commit of Git with First Repository Setup.

Aim of the Experiment:

The aim of this experiment is to install, initialize, configure, and perform the first commit using Git while setting up a version-controlled repository. This experiment helps in understanding the fundamental concepts of Git, including repository creation, staging, committing changes, and tracking version history, which are essential for efficient source code management and collaboration in software development.

Objective:

The objective of this experiment is to:

- Install Git on the system.
- Configure Git with a username and email.
- Initialize a new repository.
- Add and commit files to version control.
- Understand basic Git commands for managing a repository.

Theory:

Introduction to Git:

Git is a **distributed version control system (DVCS)** used for tracking changes in files and collaborating on projects. It allows multiple developers to work on the same codebase without conflicts.

Features of Git:

- Tracks changes in source code.
- Allows multiple contributors.
- Provides version history and rollback options.
- Enables branching and merging of code.
- Works locally and integrates with remote repositories like GitHub.

Key Concepts:

- **Repository:** A collection of files and folders managed by Git.
- **Commit:** A saved change in Git.
- **Staging Area:** A place where files are prepared for a commit.
- **Branch:** A separate version of a repository.

Requirements:

- A computer with Windows, Linux, or macOS.
- Internet connection (for Git installation and remote repository setup).
- A GitHub account (optional, for remote repository).

Procedure:

Step 1: Install Git

For Windows:

1. Download Git from Git for Windows.
2. Run the installer and keep default settings.
3. Open Git Bash and verify installation

```
sh  
  
git --version
```

Copy Edit

Expected Output:

```
nginx  
  
git version 2.x.x
```

Copy Edit

For Linux (Ubuntu/Debian):

```
sh  
  
sudo apt update  
sudo apt install git -y  
git --version
```

Copy Edit

For macOS (Using Homebrew):

```
sh  
  
brew install git  
git --version
```

Copy Edit



Step 2: Configure Git

Before using Git, set up your user identity:

```
sh  
  
git config --global user.name "Your Name"  
git config --global user.email "your-email@example.com"
```

Copy Edit

Verify configuration:

```
sh  
  
git config --list
```

Copy Edit

Expected Output:

```
ini  
  
user.name=Your Name  
user.email=your-email@example.com
```

Copy Edit

Step 3: Initialize a Git Repository

1. Create a new directory for the project:

```
sh
mkdir my-first-repo
cd my-first-repo
```

Copy

Edit

2. Initialize Git in the directory:

```
sh
git init
```

Copy

Edit

Expected Output:

```
sql
Initialized empty Git repository in /path/to/my-first-repo/.git/
```

Copy

Edit

Step 4: Create and Add Files to Git

1. Create a new file:

```
sh
echo "Hello, Git!" > README.md
```

Copy

Edit

2. Check the repository status:

```
sh
git status
```

Copy

Edit

Expected Output:

```
yaml
Untracked files:
  README.md
```

Copy

Edit

3. Add the file to the staging area:

```
sh
git add README.md
```

Copy

Edit

4. Verify the status again:

```
sh
git status
```

Copy

Edit

Expected Output:

```
vbnet
Changes to be committed:
  new file:   README.md
```

Copy

Edit



Step 5: Commit the First File

1. Commit the file with a message:

```
sh                                                                    Copy Edit

git commit -m "Initial commit"
```

Expected Output:

```
sql                                                                    Copy Edit

[main (root-commit) a1b2c3d] Initial commit
1 file changed, 1 insertion(+)
create mode 100644 README.md
```

Step 6: Connect to GitHub and Push the Repository

1. Create a new repository on [GitHub](#).
2. Link the local repository to GitHub:

```
sh                                                                    Copy Edit

git remote add origin https://github.com/your-username/my-first-repo.git
```

3. Verify the remote URL:

```
sh                                                                    Copy Edit

git remote -v
```

4. Push the repository to GitHub:

```
sh                                                                    Copy Edit

git branch -M main
git push -u origin main
```

Observations:

- Git was successfully installed and configured.
- A repository was initialized and a file was added and committed.
- The basic Git workflow (add, commit, push) was followed.
- The repository was optionally connected to GitHub for remote storage.

Conclusion:

This experiment demonstrated the installation, configuration, initialization, and first commit using Git. The hands-on practice helped understand how Git tracks changes and maintains version history efficiently.

Viva Questions

1. What is Git and why is it used?
2. What is the difference between git init and git clone?
3. How do you check the status of your repository?
4. What is the purpose of git add and git commit?
5. How can you push changes to a remote repository?
6. What is the purpose of installing Git?
7. How do you verify that Git is installed on your system?
8. What is the command to configure a username in Git?
9. Why is it important to set up a username and email in Git?
10. How can you check the configured username and email in Git?
11. What does the git init command do?
12. Where is the .git folder created after initializing a repository?
13. What is the significance of the .git directory?
14. Can we initialize Git in an existing project? If yes, how?
15. What is the difference between a local repository and a remote repository?
16. What happens when you run the git add command?
17. How do you check the status of your Git repository?
18. What is the purpose of the git commit command?
19. Why should a commit message be meaningful?
20. How do you view the history of commits in Git?
21. What is the difference between tracked and untracked files?
22. How do you remove a file from staging in Git?
23. What does the git status command display?
24. How do you undo the last commit in Git?
25. What happens if you initialize Git in a folder that already has Git initialized?
26. How do you connect a local Git repository to a remote GitHub repository?
27. What is the command to push changes to a remote repository?
28. What is the purpose of git remote -v?
29. How do you rename a branch in Git?
30. How do you check the current branch in your Git repository?

Answer for Viva Questions

Here are the **answers** to all the **Viva Questions** related to Git:

1. What is Git and why is it used?

Answer: Git is a **distributed version control system** used to track changes in code, collaborate with multiple developers, and manage different versions of a project efficiently. It helps in maintaining a history of modifications and facilitates teamwork.

2. What is the difference between git init and git clone?

Answer:

- git init: Initializes a new **empty Git repository** in a directory.
- git clone: Creates a **copy of an existing repository** from a remote source like GitHub.

3. How do you check the status of your repository?

Answer: Run the following command:

```
git status
```

It shows the current state of the repository, including staged, unstaged, and untracked files.

4. What is the purpose of git add and git commit?

Answer:

- git add: Moves changes from the working directory to the staging area.
- git commit: Saves the changes in the local repository with a message.

5. How can you push changes to a remote repository?

Answer: Use the following command:

```
git push origin main
```

This sends the committed changes from the local repository to the remote repository on GitHub.

6. What is the purpose of installing Git?

Answer: Installing Git allows you to use version control, manage code history, collaborate with developers, and track changes efficiently.

7. How do you verify that Git is installed on your system?

Answer: Run:

```
git --version
```

If Git is installed, it will display the version number.

8. What is the command to configure a username in Git?

Answer:

```
git config --global user.name "Your Name"
```

9. Why is it important to set up a username and email in Git?

Answer: Git tracks contributions based on username and email. Setting them ensures that commits are properly attributed.

10. How can you check the configured username and email in Git?

Answer:

`git config --list`

or check individually:

`git config --global user.name`

`git config --global user.email`

11. What does the git init command do?

Answer: It initializes a new Git repository in a directory by creating a `.git` folder.

12. Where is the .git folder created after initializing a repository?

Answer: Inside the directory where `git init` was executed.

13. What is the significance of the .git directory?

Answer: It contains all the metadata, including commit history, branches, and configuration files for the repository.

14. Can we initialize Git in an existing project? If yes, how?

Answer: Yes, by navigating to the project folder and running:

`git init`

15. What is the difference between a local repository and a remote repository?

Answer:

- **Local Repository:** Stored on your computer.
- **Remote Repository:** Hosted on platforms like GitHub, GitLab, or Bitbucket.

16. What happens when you run the git add command?

Answer: It moves changes from the **working directory** to the **staging area**, preparing them for commit.

17. How do you check the status of your Git repository?

Answer:

`git status`

It displays changes, including staged, unstaged, and untracked files.

18. What is the purpose of the git commit command?

Answer: It saves the staged changes to the local repository with a descriptive message.

19. Why should a commit message be meaningful?

Answer: A meaningful commit message helps developers understand the purpose of the changes.

20. How do you view the history of commits in Git?

Answer:

`git log`

For a simplified version:

`git log --oneline`

21. What is the difference between tracked and untracked files?

Answer:

- **Tracked Files:** Already added to Git and being monitored for changes.
- **Untracked Files:** Not yet added to Git.

22. How do you remove a file from staging in Git?

Answer:

`git reset HEAD <file>`

23. What does the git status command display?

Answer: It shows the current state of the working directory, staged files, and untracked files.

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

Answer:

`git reset --soft HEAD~1`

This keeps the changes but removes the commit.

25. What happens if you initialize Git in a folder that already has Git initialized?

Answer: Nothing changes; the repository remains the same.

26. How do you connect a local Git repository to a remote GitHub repository?

Answer:

`git remote add origin https://github.com/your-username/repository.git`

27. What is the command to push changes to a remote repository?

Answer:

`git push -u origin main`

28. What is the purpose of git remote -v?

Answer: It lists the remote repositories connected to your local repository.

29. How do you rename a branch in Git?

Answer:

`git branch -m old-branch-name new-branch-name`

30. How do you check the current branch in your Git repository?

Answer: `git branch` or `git rev-parse --abbrev-ref HEAD`