



DevOps Internship – Task 2

Git & GitHub Fundamentals – Proof of Concept (PoC)



Objective

The objective of this task is to understand and practice **Git and GitHub fundamentals**, including version control concepts, repository management, branching, merging, and rollback operations. This PoC demonstrates a **real-world Git workflow** used in DevOps environments.



Tools Used

- **Git** – Distributed Version Control System
 - **GitHub** – Remote repository hosting platform
 - **Operating System** – Ubuntu Linux (Terminal)
-



Task Structure

DevOps-Task-2-Git-GitHub/

```
| — README.md
| — file1.txt
| — file2.txt
| — screenshots/
|   | — git-status.png
|   | — git-commit.png
|   | — git-branch.png
|   | — git-push.png
```

◆ Step-by-Step Implementation

1 Install Git & Configure User

```
git --version  
git config --global user.name "Your Name"  
git config --global user.email "yourmail@example.com"
```

✓ Ensures Git is installed and user identity is configured globally.

2 Initialize Local Git Repository

```
mkdir DevOps-Task-2-Git-GitHub  
  
cd DevOps-Task-2-Git-GitHub  
  
git init
```

📁 A hidden `.git` directory is created, which stores all version history and metadata.

3 Create Files & Track Changes

```
touch file1.txt file2.txt  
  
git status  
  
git add .  
  
git commit -m "Initial commit with sample files"
```


✓ Files are now tracked and committed to the local repository.

4 Create GitHub Repository & Link Remote

```
git remote add origin https://github.com/username/DevOps-Task-2-Git-GitHub.git
```

```
git branch -M main
```

```
git push -u origin main
```

 Local repository is now connected to GitHub.

5 Branching & Merging

```
git checkout -b feature-branch
```


```
echo "New Feature" >> file1.txt
```

```
git add file1.txt
```

```
git commit -m "Added new feature"
```

```
git checkout main
```

```
git merge feature-branch
```


 Demonstrates safe development using branches and merging into **main**.

6 Rollback & History

```
git log
```

```
git reset --soft HEAD~1
```

```
git checkout file1.txt
```

 Shows how to view commit history and undo changes safely.

