St. Francis Institute of Technology, Mumbai-400 103

**Department Of Information Technology**

A.Y. 2024-2025

Class: TE-ITA/B, Semester: V

Subject: **DevOps Lab**

**Experiment – 2: To understand version control system/ source code management, install git and create a GitHub account**

1. **Aim:** To understand version control using Git and create a GitHub account
2. **Objectives:** Aim of this experiment is that, the students will be able

* To be aware of different Version Control tools like GIT and GitHub
* To obtain complete knowledge of the “version control system” to effectively track changes augmented with Git and GitHub

1. **Outcomes:** After study of this experiment, the students will be able to

* GIT Installation
* Version Control
* Working with remote repository

1. **Prerequisite:** Knowledge of software engineering concept of version control
2. **Requirements:** Git,Personal Computer, Windows operating system, browser, Internet Connection, Microsoft Word.
3. **Pre-Experiment Exercise:**

**Brief Theory:** Refer shared material

1. **Laboratory Exercise**
   * + 1. **Procedure:**

**a. Answer the following:**

* Explain version control and its types
* What is Git and GitHub?
* Explain different Git commands.

**b**. **Execute following on Git and GitHub (Refer the shared material) and attach screenshots:**

* Git installation
* Git commands
* GitHub account creation

1. **Post-Experiments Exercise**
2. **Extended Theory:**

Nil

1. **Questions:**

* What are the different Git areas? Explain with diagram
* What is a Git conflict?

1. **Conclusion:**

* Write what was performed in the experiment.
* Write the significance of the topic studied in the experiment.

1. **References:**

[**https://github.com/**](https://github.com/)

[**https://guides.github.com/activities/hello-world/**](https://guides.github.com/activities/hello-world/)

[**https://git-scm.com/docs/gittutorial**](https://git-scm.com/docs/gittutorial)

**Laboratory Exercise:**

1. **Answer the following:**
2. **Explain version control and its types.**

Version control is a system that records changes to a file or set of files over time so that specific versions can be recalled later. It helps manage changes in documents, computer programs, large websites, and other collections of information.

Types of Version Control Systems:

1. **Local Version Control Systems:**

Keep track of files within the local file system. **Example:** RCS (Revision Control System).

1. **Centralized Version Control Systems (CVCS):**

Use a central server to store all versions of a project. Clients check out files from the central repository. **Examples:** Subversion (SVN), CVS (Concurrent Versions System).

1. **Distributed Version Control Systems (DVCS):**

Every contributor has a local copy of the entire project history. Changes are tracked independently and can be merged later. **Examples:** Git, Mercurial, Bazaar.

**2. What is Git and GitHub?**

**Git:** Git is a distributed version control system designed to handle everything from small to very large projects with speed and efficiency. It allows multiple people to work on the same project independently and concurrently, tracking changes and enabling the merging of contributions.

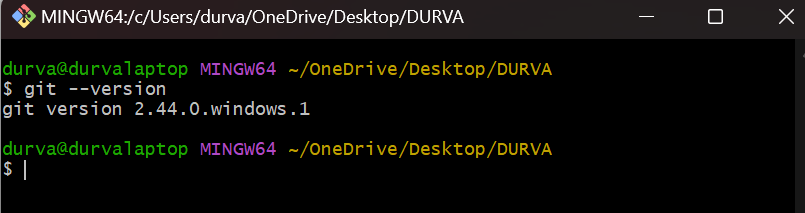
**GitHub:** GitHub is a web-based platform that uses Git for version control and offers additional features such as bug tracking, feature requests, task management, and wikis for every project. It facilitates collaboration and sharing of code among developers.

**3. Explain different Git commands.**

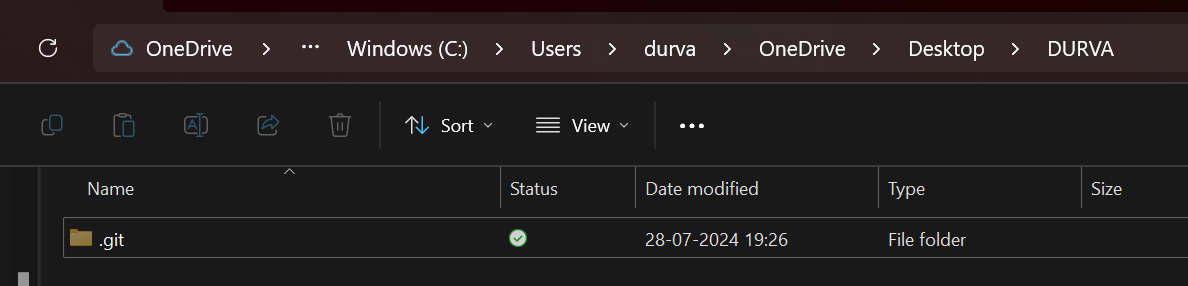
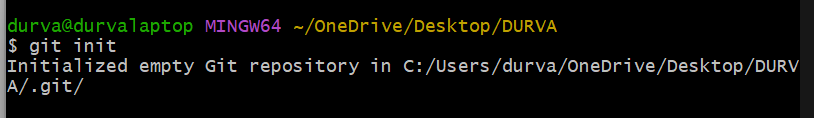
1. **git init**: Initialize a repository. git init
2. **git clone**: Clone a repository. git clone <repo\_url>
3. **git add**: Add files to staging. git add <file> or git add .
4. **git commit**: Commit changes. git commit -m "message"
5. **git status**: Show status. git status
6. **git push**: Push changes. git push <remote> <branch>
7. **git pull**: Pull changes. git pull <remote> <branch>
8. **git branch**: Manage branches. git branch (list), git branch <name> (create), git branch -d <name> (delete)
9. **git checkout**: Switch branches. git checkout <branch> or git checkout -b <new\_branch>
10. git merge: Merge branches. git merge <branch>
11. **git log**: Show commit logs. git log
12. **git remote**: Manage remotes. git remote -v (list), git remote add <name> <url> (add)

**B. Execute following on Git and GitHub (Refer the shared material) and attach screenshots:**

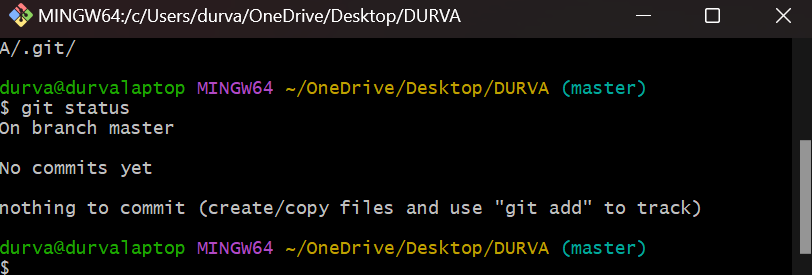
**GIT VERSION:**



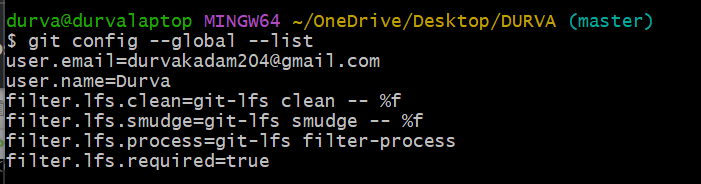
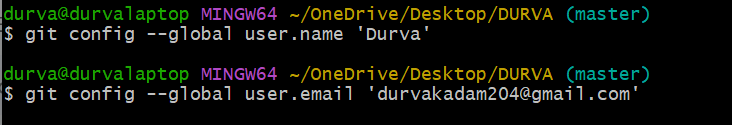
**GIT INIT:**



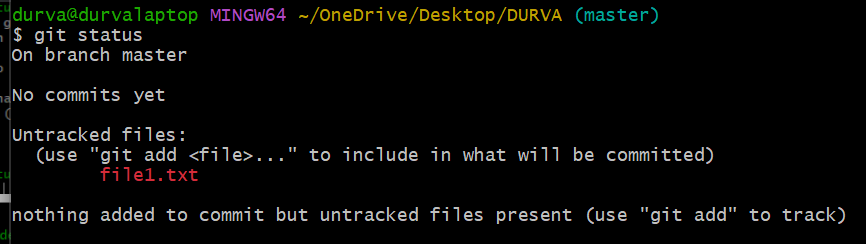
**GIT STATUS:**



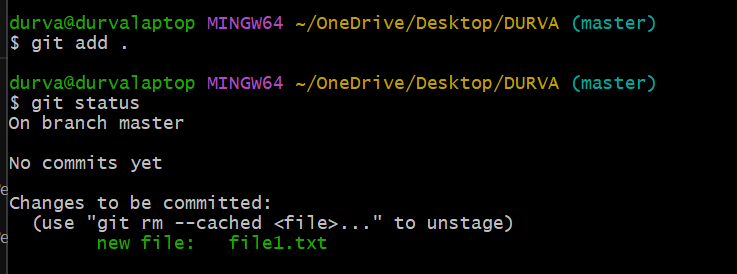
**GIT CONFIGURATION:**



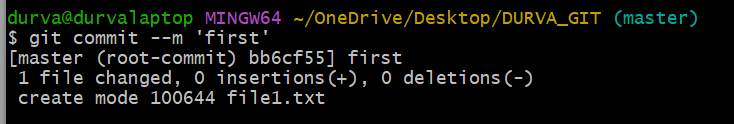
**GIT STATUS:**

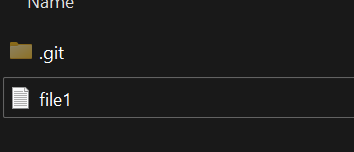


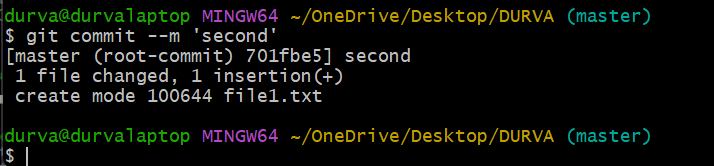
**GIT ADD:**



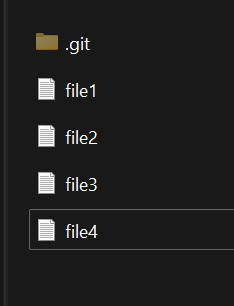
**GIT COMMIT:**

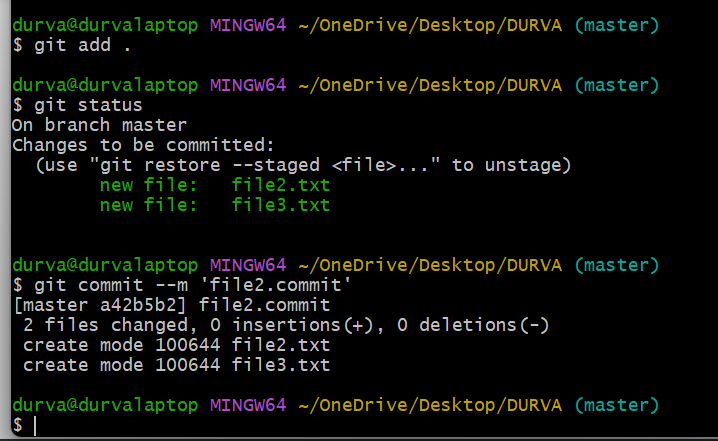


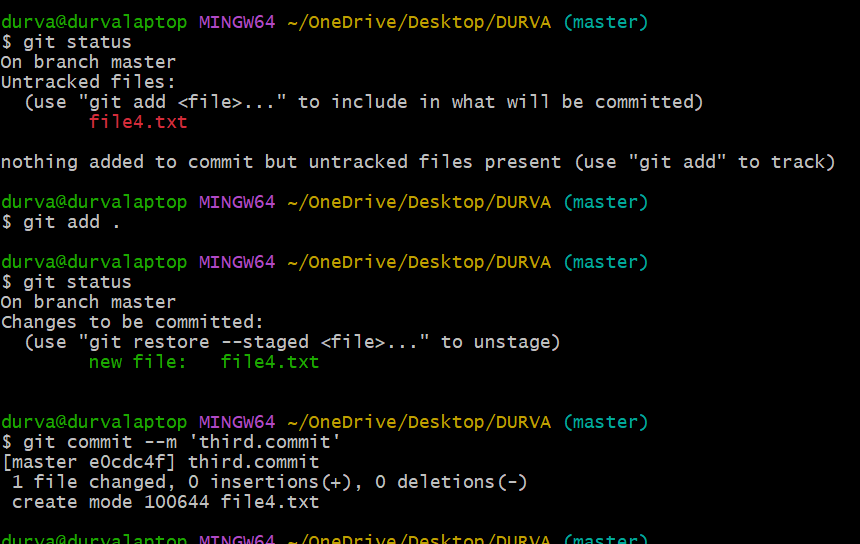


**MORE GIT COMMITS:**

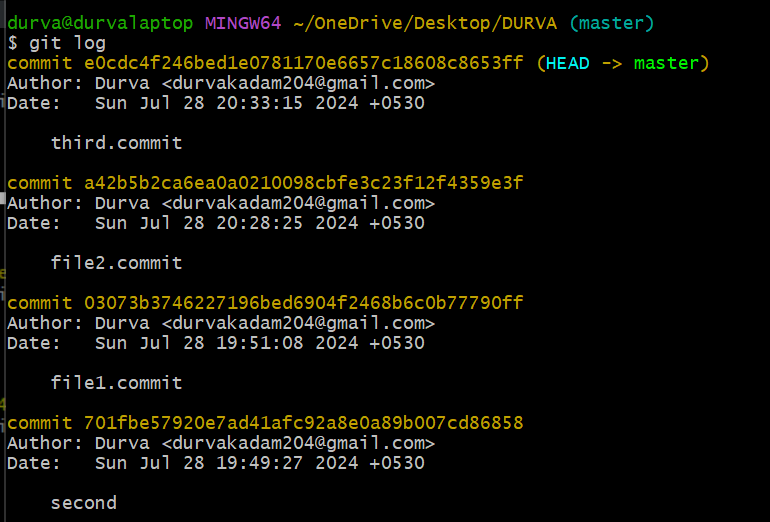
**GIT COMMIT (MULTIPLE FILES):**

****

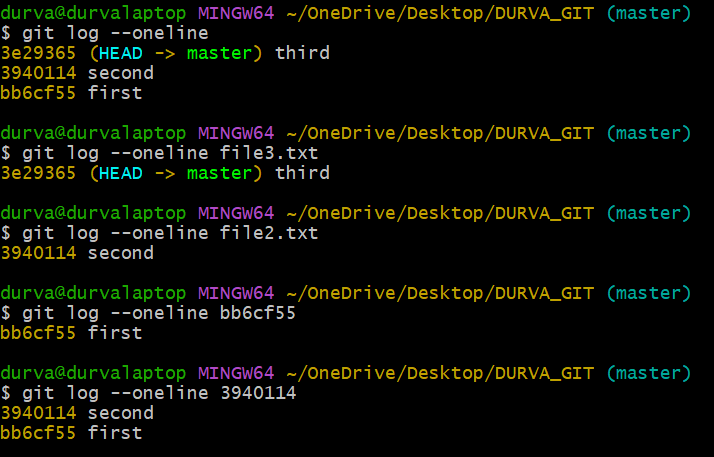




**GIT LOG:**



**GIT LOG (VARIATIONS):**

****

**GITHUB ACCOUNT :**

