2025

VERSION CONTROL : GIT, GIT BASH, GIT HUB

NADEEM SIDDIQI

LOCAL & REMOTE GIT PROJECT
VM-PRO-NIT-0001
5/8/2025

GIT/GIT Bash/GIT HUB

Version Control

PREPARED BY:

NIT ACADEMY

NADEEM M. SIDDIQI

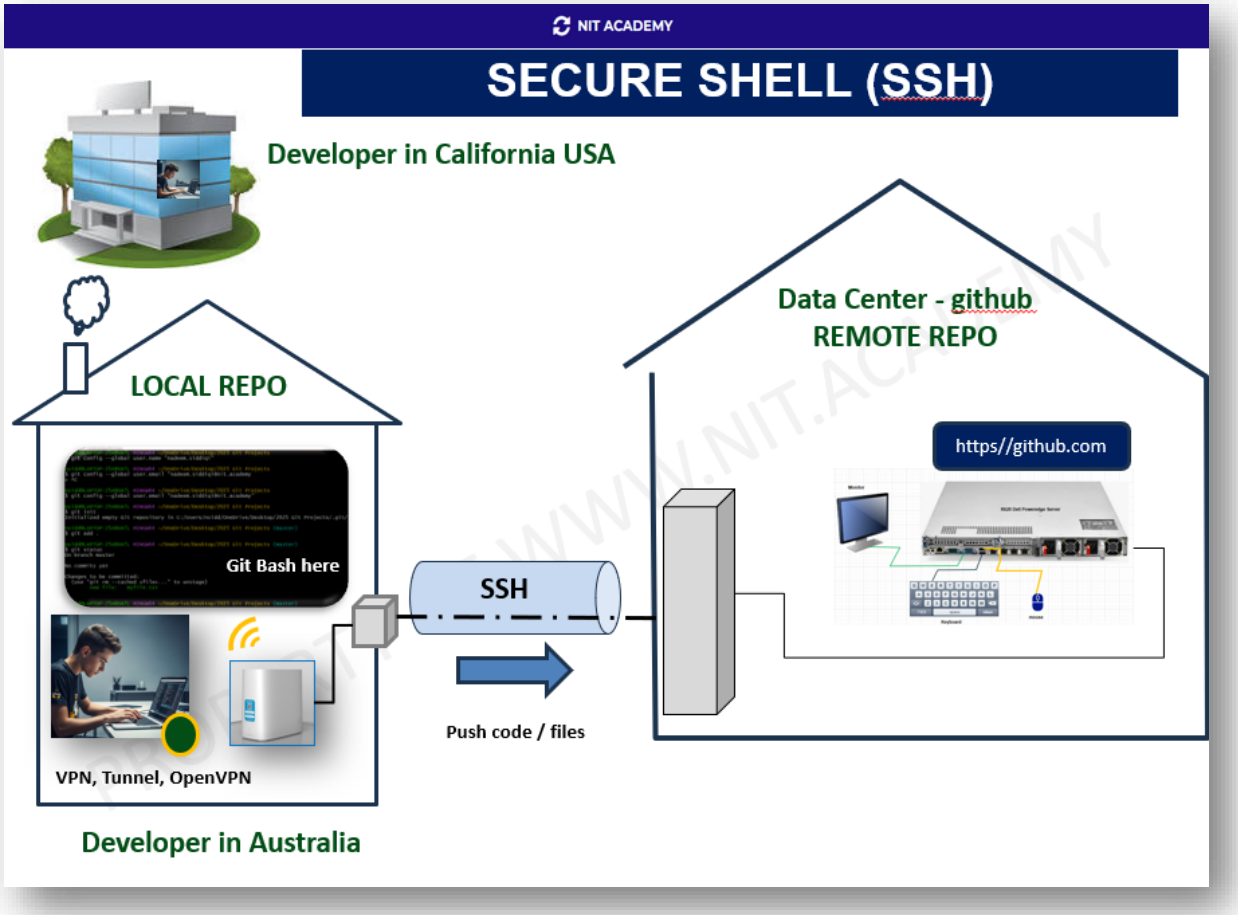
01	AUG-05-2025	Issued for Review	NIT	KAMRAN	NADEEM
Rev.	**Date**	**Reason for Issue**	**Prepared by**	**Checked by**	**Approved by**

~ Page left Blank Intentionally ~

Table of Contents

2.0 INTRODUCTION

When developers are sitting in all different parts of the world working on a project collaboratively, they all need a method whereby they can share their **"bit"** of work with **others**. In the picture below, two developers are working from different locations so that need a way to control their documents. GIT provides version control. We will learn how to create a local storage (called local repo) and then push it to a central location that is remotely located called "github".



3.0 WHAT IS GIT?

Git is a **version control system** (VCS). It helps you:

- **Track changes** in your source code and/or documentation.
- **Collaborate** with others without overwriting each other's work.
- **Go back in time** to previous versions if needed.

Practical application

- Save your project at different stages (commits).
- Revert to an earlier stage if something breaks.
- Work on multiple features in parallel using **branches.**

4.0 WHAT IS GITHUB?

GitHub is a **cloud-based hosting service for Git repositories**. It allows you to:

- **Store your code / documentation online**
- **Collaborate** with others (open source or private teams)
- **Use tools** like pull requests, issues, CI/CD, etc.

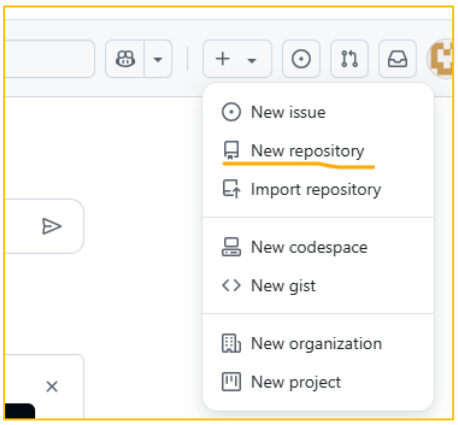
4.1 Opening a GitHub account Online

STEP 1: Click on https://github.com/ and follow the following video

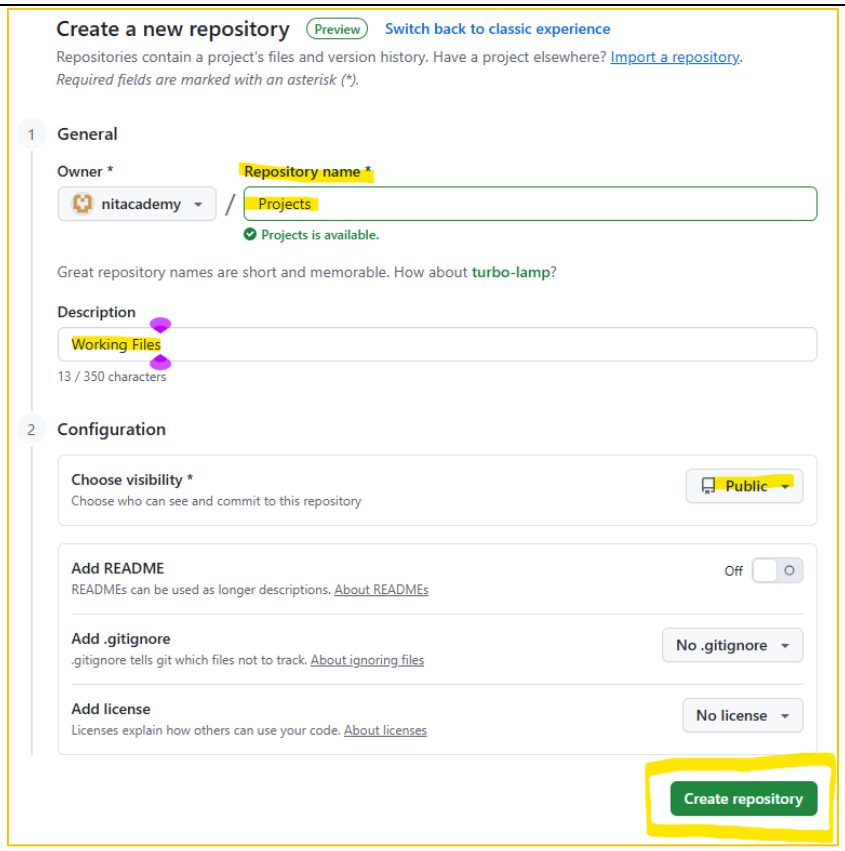
https://www.youtube.com/watch?v=Gn3w1UvTx0A

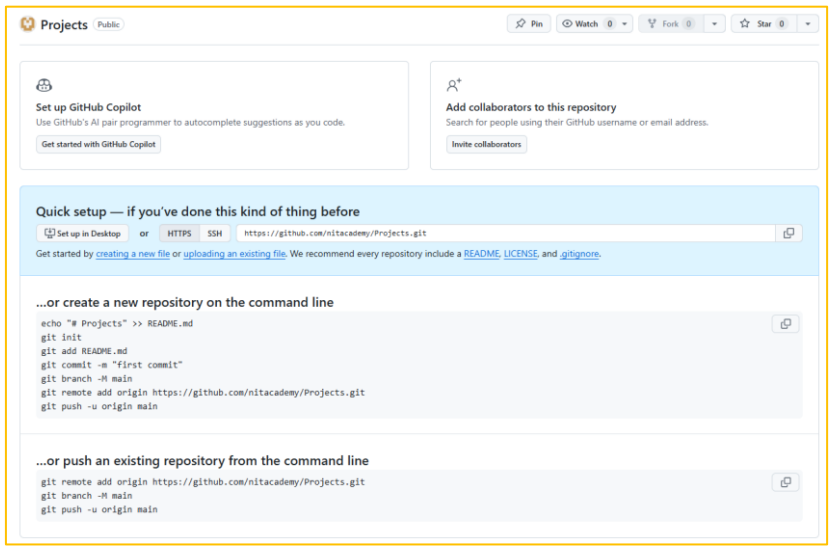
4.2 Creating our first REMOTE Storage (Repository) on GitHub

- Log in to GitHub.
- Left Click on the "+" sign (usually in the top-right corner or on your dashboard).
- Click **New repository.**



- Give it a **name** Projects
- Choose **Public** or **Private**.
- **Do NOT** check "Initialize with a README" if your local repo already has files — otherwise you might have merge conflicts.
- Click **Create repository**.


Create a new repository
Preview
Switch back to classic experience
Repositories contain a project's files and version history. Have a project elsewhere? Import a repository.
Required fields are marked with an asterisk (*).
1 General
Owner *
Repository name *
nitacademy
Projects
Projects is available.
Great repository names are short and memorable. How about turbo-lamp?
Description
Working Files
13 / 350 characters
2 Configuration
Choose visibility *
Choose who can see and commit to this repository
Public
Add README
READMEs can be used as longer descriptions. About READMEs
Off
Add .gitignore
.gitignore tells git which files not to track. About ignoring files
No .gitignore
Add license
Licenses explain how others can use your code. About licenses
No license
Create repository

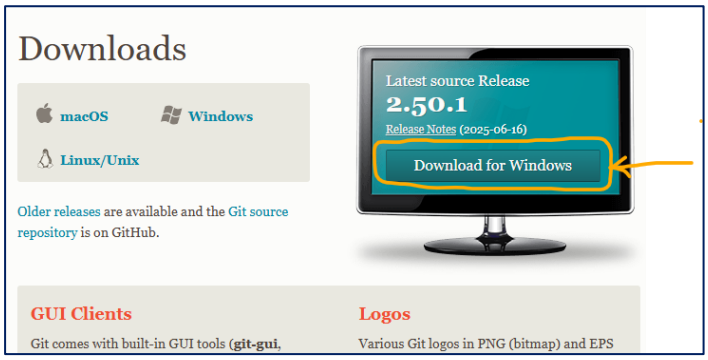

Projects
Public
Pin
Watch 0
Fork 0
Star 0
Set up GitHub Copilot
Use GitHub's AI pair programmer to autocomplete suggestions as you code.
Get started with GitHub Copilot
Add collaborators to this repository
Search for people using their GitHub username or email address.
Invite collaborators
Quick setup — if you've done this kind of thing before
Set up in Desktop
or
HTTPS
SSH
https://github.com/nitacademy/Projects.git
Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.
...or create a new repository on the command line
echo "# Projects" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/nitacademy/Projects.git
git push -u origin main
...or push an existing repository from the command line
git remote add origin https://github.com/nitacademy/Projects.git
git branch -M main
git push -u origin main

5.0 WHAT IS GIT BASH?

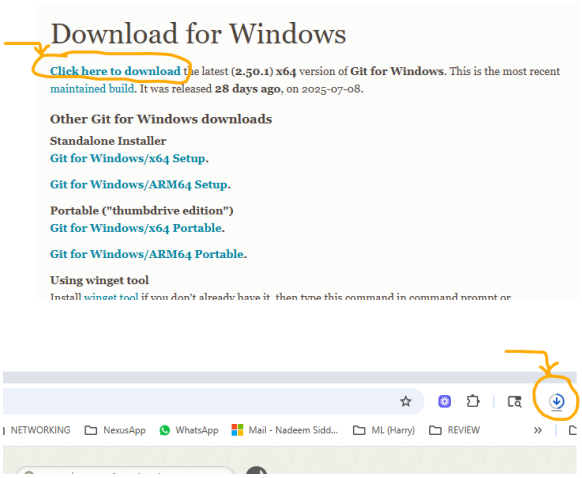
Git Bash is a **command-line application for Windows** that lets you use Git commands and Unix/Linux-like commands in Windows. Follow the following steps to Install "git-bash" on your Windows or macOS

STEP 1: https://git-scm.com/downloads

STEP 2: Click on "Download for Windows" or "macOS" for mac users



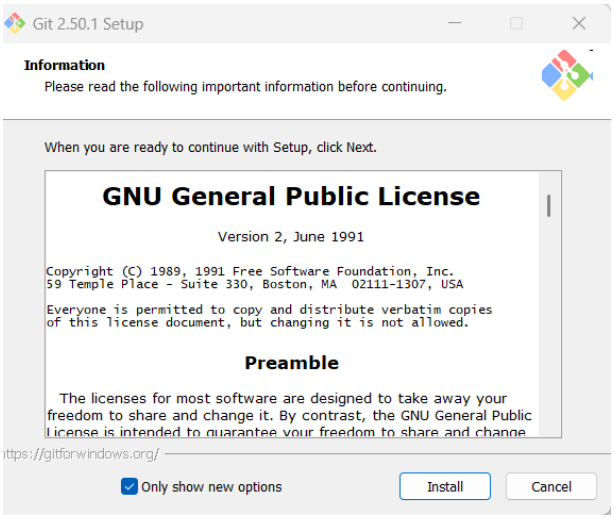
STEP 3:



STEP 4: Check for the downloaded file

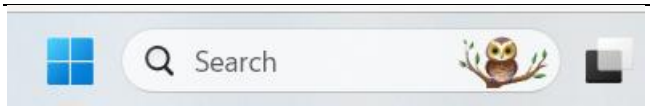
Windows will show a pop-up window: "Do you want to allow this app to make changes to your device" click "**yes**"

Then the following window will appear. Click **"Install" and** just continue clicking on **"next"** until installation starts. Download **takes time** so don't panic!

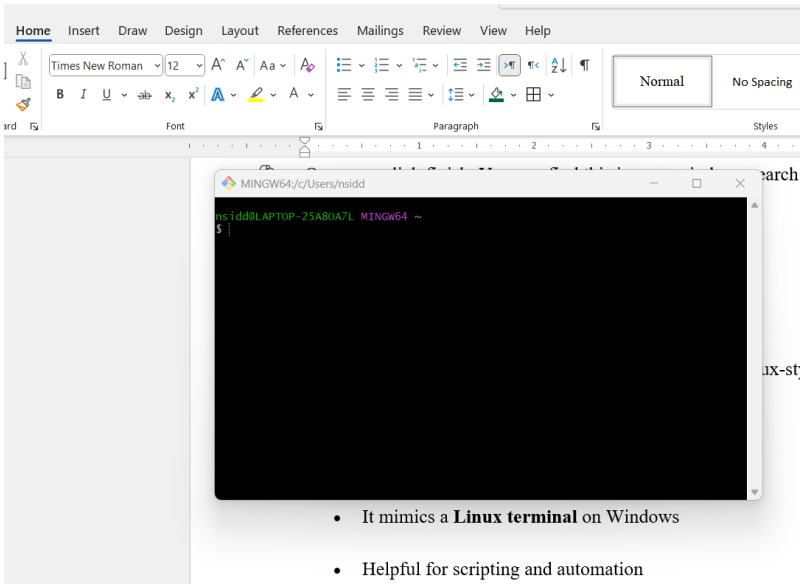


STEP 5: Click on "Launch Git Bash" and then Finish. What is Git Bash? This is a terminal similar to Windows "Command Prompt" and "Terminal"

STEP 6: Once you click finish. You can find this in your windows Search by typing git bash



The pop-up will look similar to the window below:



- Test by running “ls” command

5.1 Why use Git Bash?

- You get **full Git command-line features**
- It mimics a **Linux terminal** on Windows
- Helpful for scripting and automation
- More **control and flexibility** than using GUI apps

6.0 SUMMARY OF WHAT TO INSTALL ON WINDOWS

Component	Purpose
Git Bash	Terminal to run Git + Linux commands
GitHub	Online platform to store/share Git repos

7.0 LAB PRACTICE

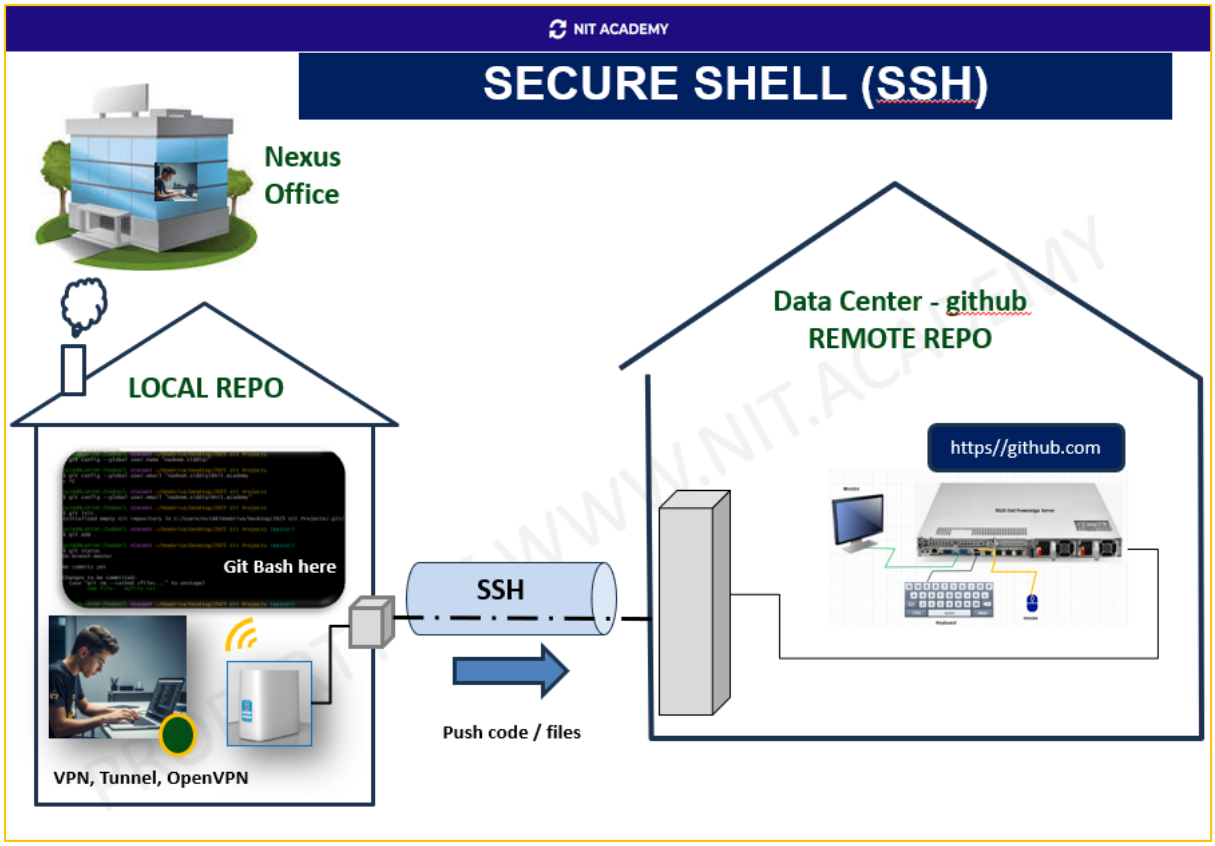


Figure 1:git in action: pushing code/files from local to remote repository

7.1 Lab Objective

Based on Figure 1 our goal is to:

1. Create a local repository
2. Use git as the version control tool to manage local and global repositories
3. PUSH code/files from our local/laptop like developers do onto the remote repository called gihub.

By the end of this lab, you will:

- Initialize a local repo on your Windows, **a Git repository**

- Track changes
- Stage the local files
- Commit
- Push your document(s) to GitHub

7.2 Creating a Repository Folder on your Windows / mac Machines:

Step 1: Create a folder on your local machine (Windows/mac)

Right-click anywhere on your desktop and create a new folder and name it **"2025 Git Projects"**

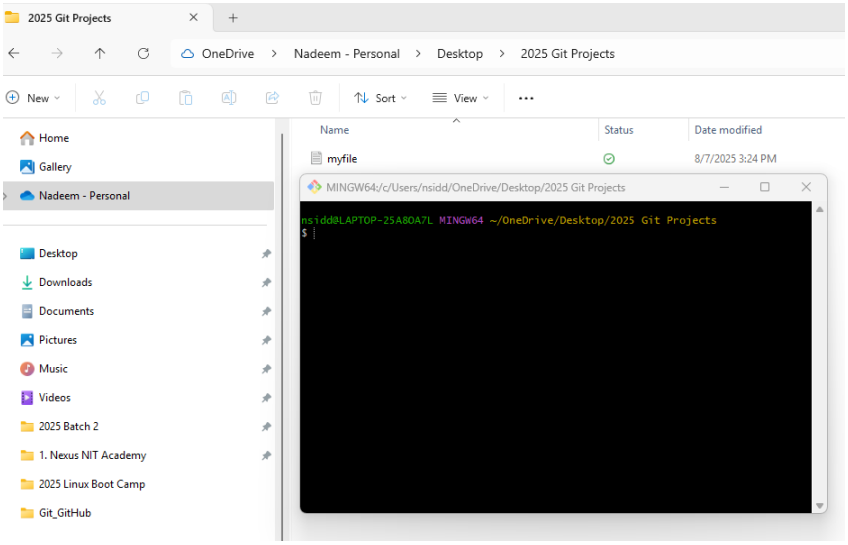
Step 2: Create a "Text Document" in the 20235 Git Projects Folder

- Open **"2025 Git Projects"** folder
- Right click and select "New" and then click on "Text Document"
- Name/Rename this document "myfile"
- Write "Testing GitHub", save and close the document

7.3 Working with Git Bash

Step 1: Open Git Bash

- Right-click anywhere inside your "2025 Git Projects" folder → select "Show more options" if you don't see **"Open Git Bash here".**
- **Select "Open Git Bash here"** => you should see the following:



Step 2: Configure Git (First-Time Only)

Update "User Name" and "User Email"

NOTE: Use "nit.academy" as your FQDN (Fully Qualified Domain Name)

- git config --global user.name "**Your Name**" => example, "patrick.hayden"
- git config --global user.email youremail@example.com => example, patrick.hayden@nit.academy

Step 3: Initialize your local repository

- $git init

Step 4: Check the status of your LOCAL repository commonly called "Repo"

- $git status

Step 5: Let us STAGE our documents/files – basically review before we send them away to github (REMOTE Repository)

- $git add . (don't forget the "." ; it means 'present working directory"
- $git status

Step 6: Removing that which is staged

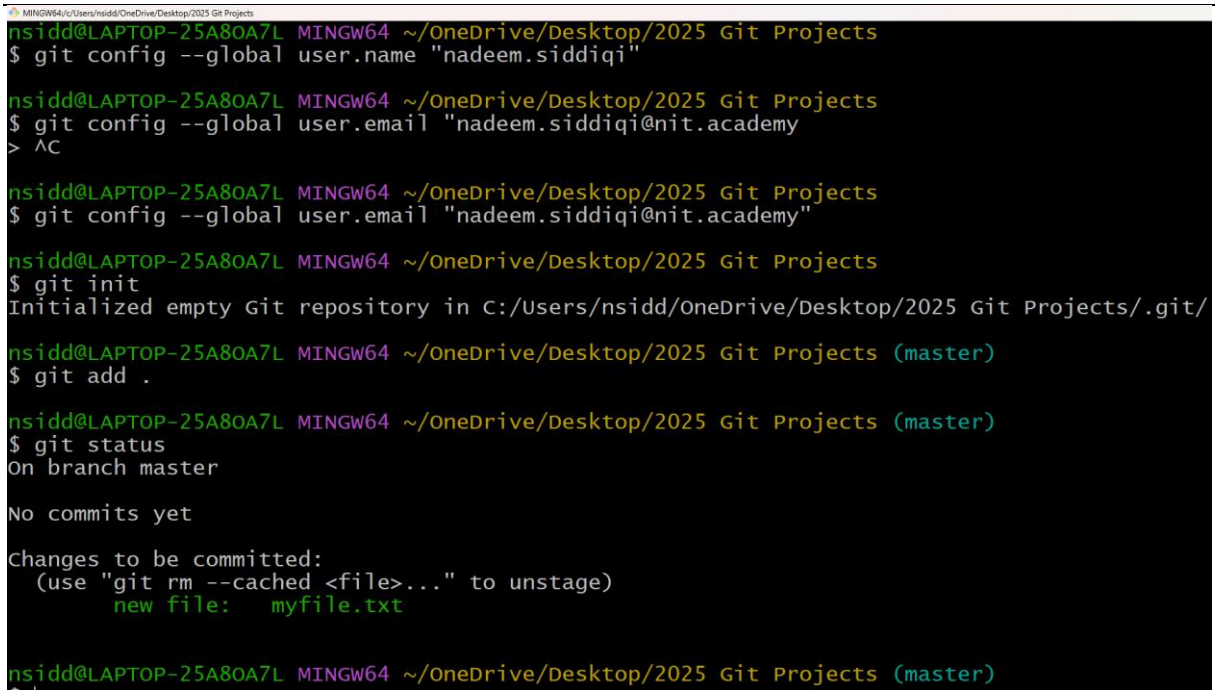
- $git rm –cached myfile.txt$git status
- $git status

Step 6: Stage it again

- $git add .
- $git status

Step 6: Once reviewed let us "save" our work that is ready to be sent to github

- $commit –m "version 1" (but we can PUSH from our local machine to github yet!)



8.0 GENERATING SSH KEYS

Follow the STEPS below to generate SSH Keys and

Step 1: on your Git bash we will generate some keys

- ssh-keygen -t rsa -b 4096 -C patrick.hayden@nit.academy

Step 2: Follow the steps below:

- $ eval "$(ssh-agent -s)"

Step 3:

- $ ssh-add ~/.ssh/id_rsa

Identity added: /c/Users/nsidd/.ssh/id_rsa (Patrick.hayden@nit.academy)

NOTE: (Replace id_rsa with your actual GitHub private key file if it's different, like id_ed25519.)

Step 4: The command below will "copy" the keys for you

clip < ~/.ssh/id_rsa.pub

Open "Notepad" and paste the key using "Cntrl+v" on your keyboard

Step 5: Let us login www.github.com your account

https://www.youtube.com/watch?v=X40b9x9BFGo

- Click on "Settings"
- Click on "SSH and GPG Keys"
- Click on "New SSH Key"

1. Add Title

2. Add Description

3. "Cntrl+v" and paste then

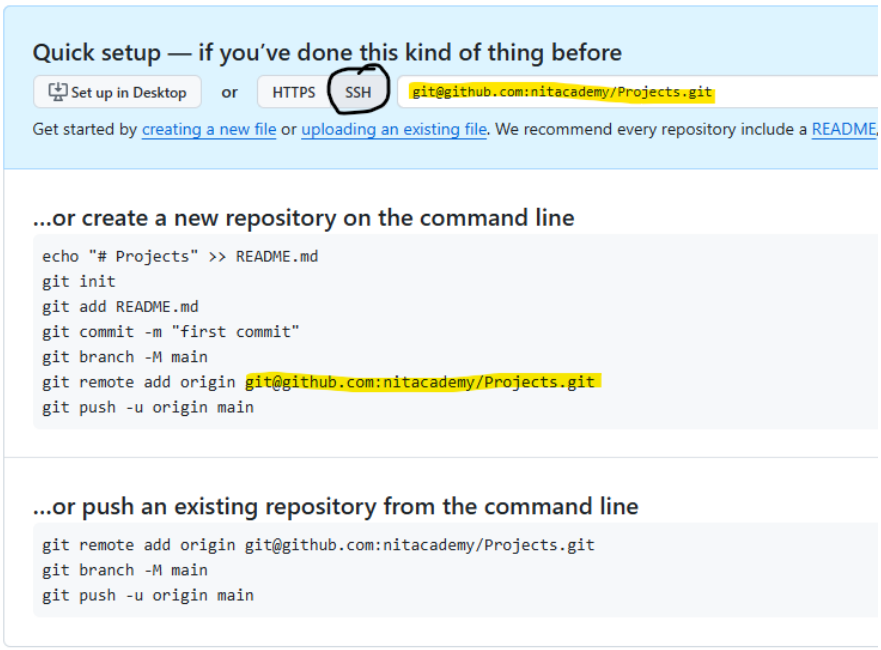
4. Finally "add ssh key"

Final Note:

You are now ready to "push" and "pull" code between your local PC and remote GitHub Repository

9.0 BACK TO LOCAL REPO!!!

From you github make sure you select “SSH” and copy the link/url as in the picture below:



Step 1: Check the name of your branch it should be "master"

$git branch

Step 2 ADD REMOTE REPO (GITHUB)

$git remote add origin git@github.com:nitacademy/Projects.git

Step 3: Push code from Local PC => Laptop (Remote Repository)

$git push -u origin master

Student Practice Questions

1. What does git init do?
2. How do you check what files are staged?
3. What is the difference between git add and git commit?
4. Why do we use git push?
5. What is the command to see your current Git settings?
6. What happens if you run git status before git add?

---

1. Optional Challenges (DO NOT PROCEED)

- Create and commit a second file (e.g., readme.md)
- Make changes to hello.txt and commit again
- Create a new branch using:

$git checkout -b test-branch