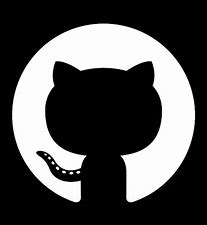
****

**BY: -**

**Abhishek Mittal**

**2110990067**

**SCM PROJECT**

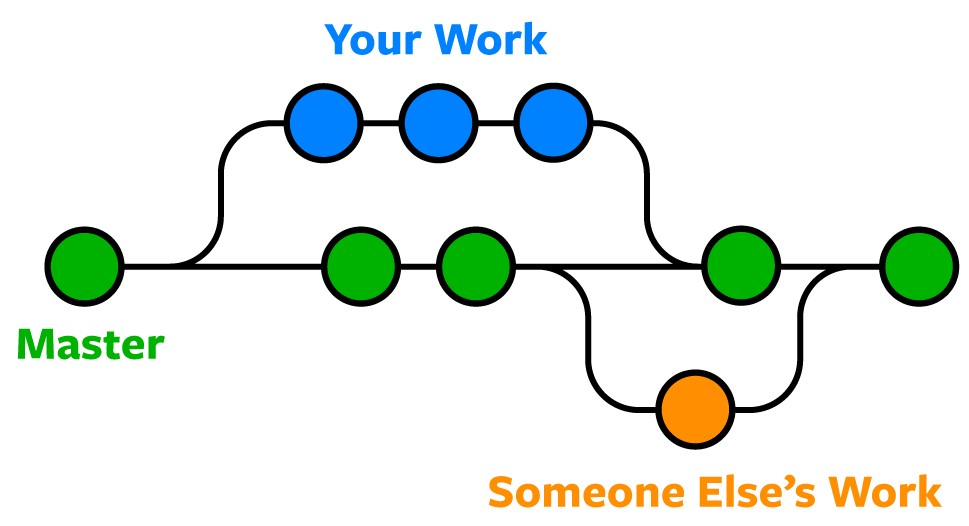
**TASK 1.1**

**What is Git?**

Git is a **specific open-source version control system** created by **Linus Torvalds** in 2005.

Specifically, Git is a **distributed version control system**, which means that the entire codebase and history is available on every developer’s computer, which allows for easy branching and merging.

It is software for tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development



**What Is GitHub?**

GitHub is a website and cloud-based service that helps developers store and manage their code, as well as track and control changes to their code. To understand exactly what GitHub is, you need to know two connected principles

**Differences between Git and GitHub:**

**Git:** Git is a distributed version control system for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files. Its goals include speed, data integrity, and support for distributed, non-linear workflows.

**GitHub:** GitHub is a web-based Git repository hosting service, which offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features.

|  |  |  |
| --- | --- | --- |
| **S. No** | **Git** | **GitHub** |
| 1 | Git is a software | GitHub is a service |
| 2 | Git is a command-line tool | GitHub is a graphical user interface |
| 3 | Git is installed locally on the system | GitHub is hosted on the web |
| 4 | Git is maintained by Linux | GitHub is maintained by Microsoft |
| 5 | Git is focused on version control and code sharing hosting | GitHub is focused on centralized source code hosting. |

**Installing and Configuring the Git client**

B

A

The following sections list the steps required to properly install and configure the Git clients - Git Bash and Git GUI - on a Windows 7 computer. Git is also available for Linux and Mac. The remaining instructions here, however, are specific to the Windows installation.

Be sure to carefully follow all of the steps in the first five sections.

The last section, 6,

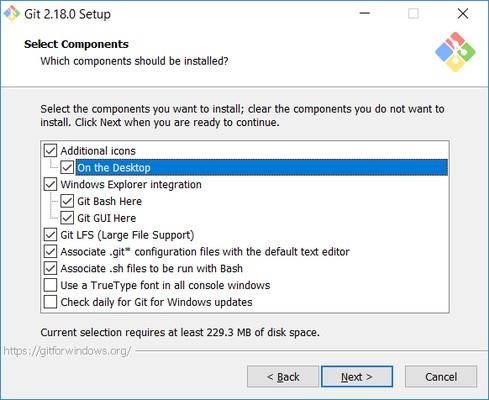
is optional. There is also a section

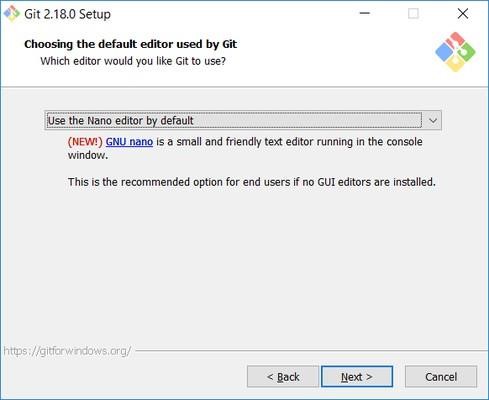
on common problems and possible fixes at the bottom of the document

# **Git installation**

Download the Git installation program (Windows, Mac, or Linux) from http://git-scm.com/downloads. When running the installer, various screens appear (Windows screens shown). Generally, you can accept the default selections, ***except in the screens below where you do NOT want the default selections:***

In the **Select Components** screen, make sure **Windows Explorer Integration** is selected as shown:





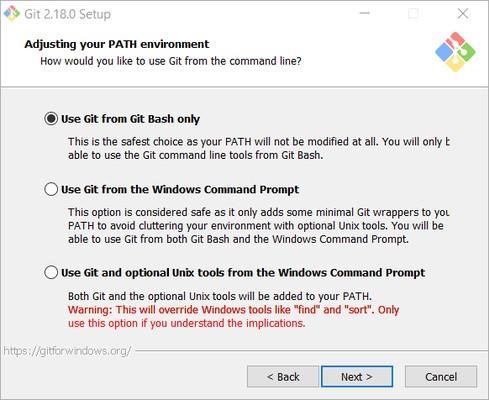
In the **Adjusting your PATH** screen, all three options are acceptable:

**Use Git from Git Bash only**: no integration, and no extra commands in your command path

**Use Git from the Windows Command Prompt**: adds flexibility - you can simply run git from a Windows command prompt, and is often the setting for people in industry - but this does add some extra commands.

**Use Git and optional Unix tools from the Windows Command Prompt**: this is also a robust choice and useful if you like to use

Unix commands like grep.



In the Configuring the line ending screen, select the middle option (Checkout as-is, commit Unix-style line endings) as shown. This helps migrate files towards the Unix-style (LF) terminators that most modern IDE's and editor’s support. The Windows convention (CR-LF line termination) is only important for Notepad (as opposed to Notepad++), but if you are using Notepad to edit your code you may need to ask your instructor for help.



**Setting up GitHub Account**

The first steps in starting with GitHub are to create an account, choose a product that fits your needs best, verify your email, set up two-factor authentication, and view your profile.

There are several types of accounts on GitHub. Every person who uses GitHub has their own user account, which can be part of multiple organizations and teams. Your user account is your identity on GitHub.com and represents you as an individual

## **1. Creating an account**

To sign up for an account on GitHub.com, navigate to<https://github.com/>and follow the prompts. To keep your GitHub account secure you should use a strong and unique password. For more information, see "[Creating a strong password.](https://docs.github.com/en/github/authenticating-to-github/keeping-your-account-and-data-secure/creating-a-strong-password)"

## **2. Choosing your GitHub product**

You can choose GitHub Free or GitHub Pro to get access to different features for your personal account. You can upgrade at any time if you are unsure at first which product you want. For more information on all of GitHub's plans, see "[GitHub's products.](https://docs.github.com/en/get-started/learning-about-github/githubs-products)"

## **3. Verifying your email address**

To ensure you can use all the features in your GitHub plan, verify your email address after signing up for a new account. For more information, see "[Verifying your email address.](https://docs.github.com/en/github/getting-started-with-github/signing-up-for-github/verifying-your-email-address)"

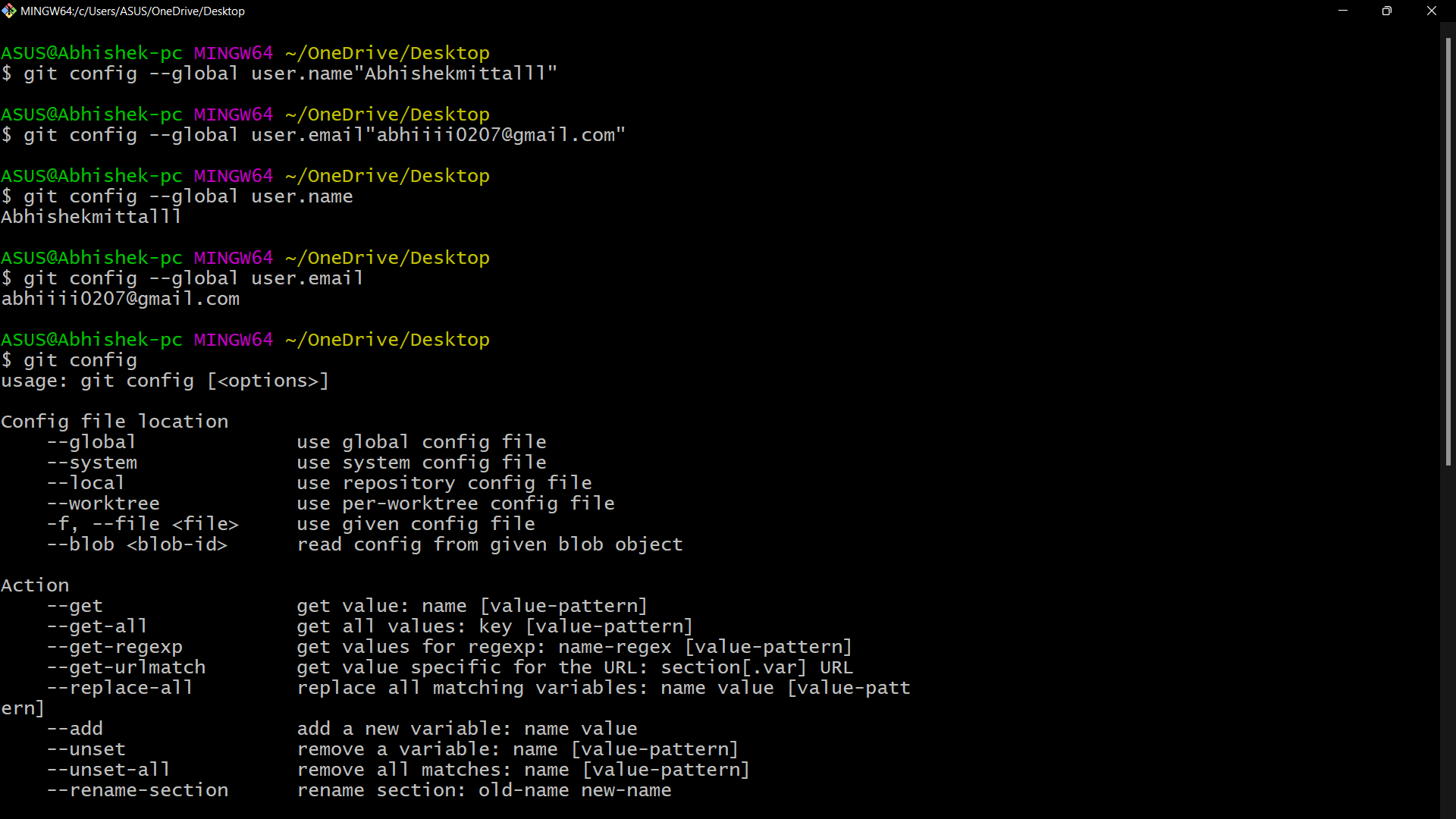
## **4. Configuring two-factor authentication**

Two-factor authentication, or 2FA, is an extra layer of security used when logging into websites or apps. We strongly urge you to configure 2FA for the safety of your account. For more information, see "[About two-factor authentication.](https://docs.github.com/en/github/authenticating-to-github/securing-your-account-with-two-factor-authentication-2fa/about-two-factor-authentication)"

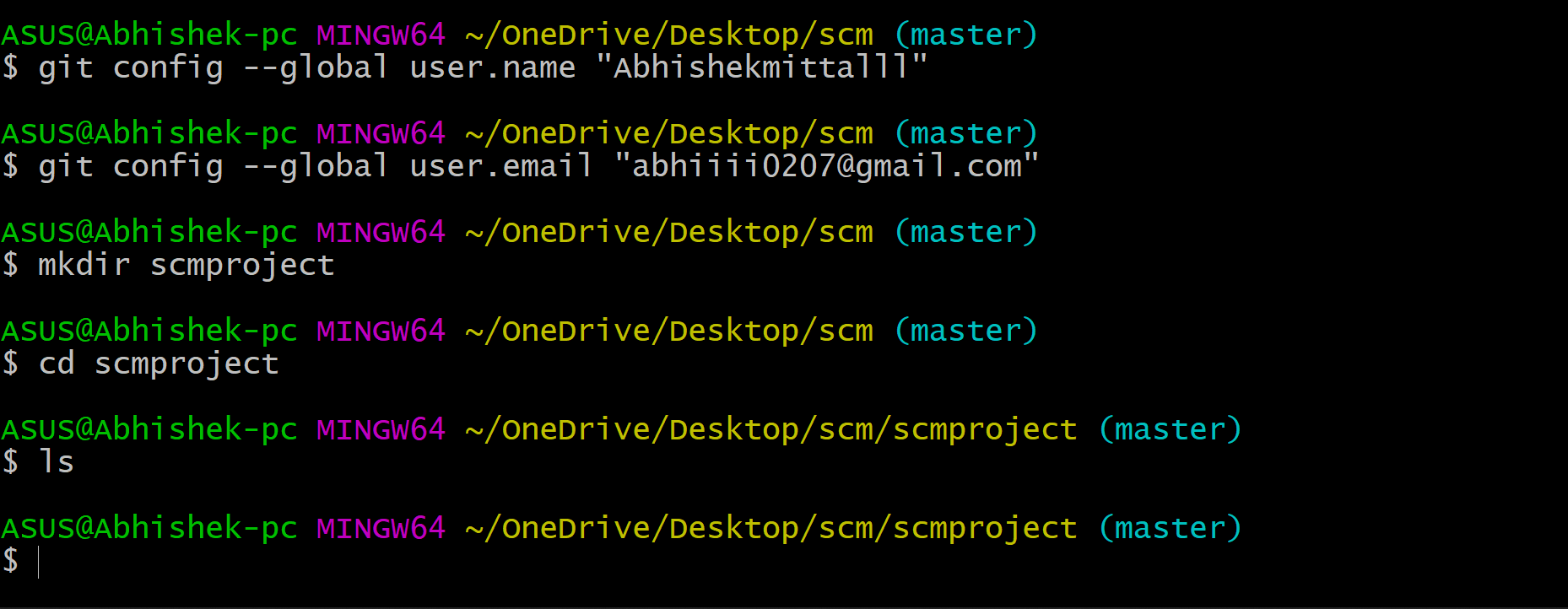
## **5. Viewing your GitHub profile and contribution graph**

Your GitHub profile tells people the story of your work through the repositories and gist’s you've pinned, the organization memberships you've chosen to publicize, the "[About your profile"](https://docs.github.com/en/github/setting-up-and-managing-your-github-profile/customizing-your-profile/about-your-profile) and "[Viewing contributions on your profile.](https://docs.github.com/en/github/setting-up-and-managing-your-github-profile/managing-contribution-graphs-on-your-profile/viewing-contributions-on-your-profile)"

**SETTING USER-NAME AND EMAIL**



**BASIC COMMAND FOR GIT- BASH**

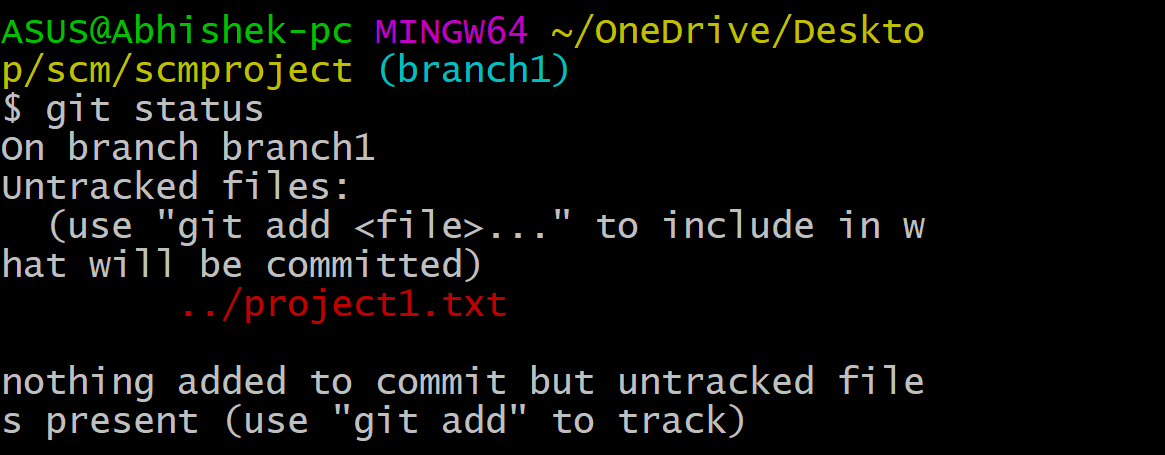
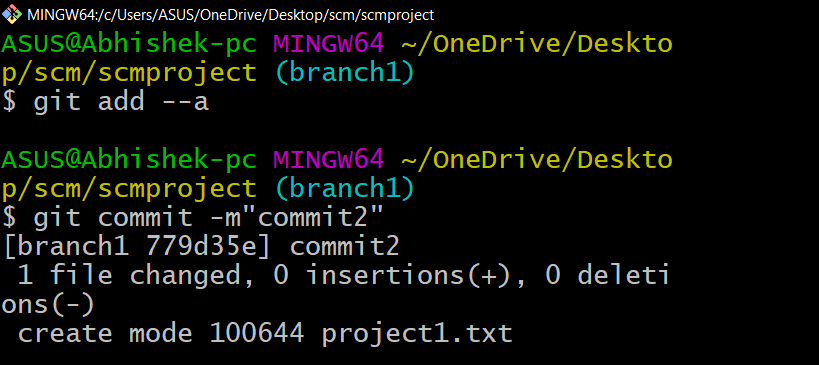
****

**mkdir**: - It is used for creating a directory using Git Bash.

**cd command:** - Both Bash and Windows console host have a cd command. cd is an acronym for 'Change Directory'. cd is invoked with an appended directory name. Executing cd will change the terminal sessions current working directory to the passed directory argument.

**Pwd**: - The Bash command pwd is used to print the 'present working directory'. This is the folder or path that the current Bash session resides in.

**ls**: -The Git bash command ls is used to 'list' contents of the current working directory

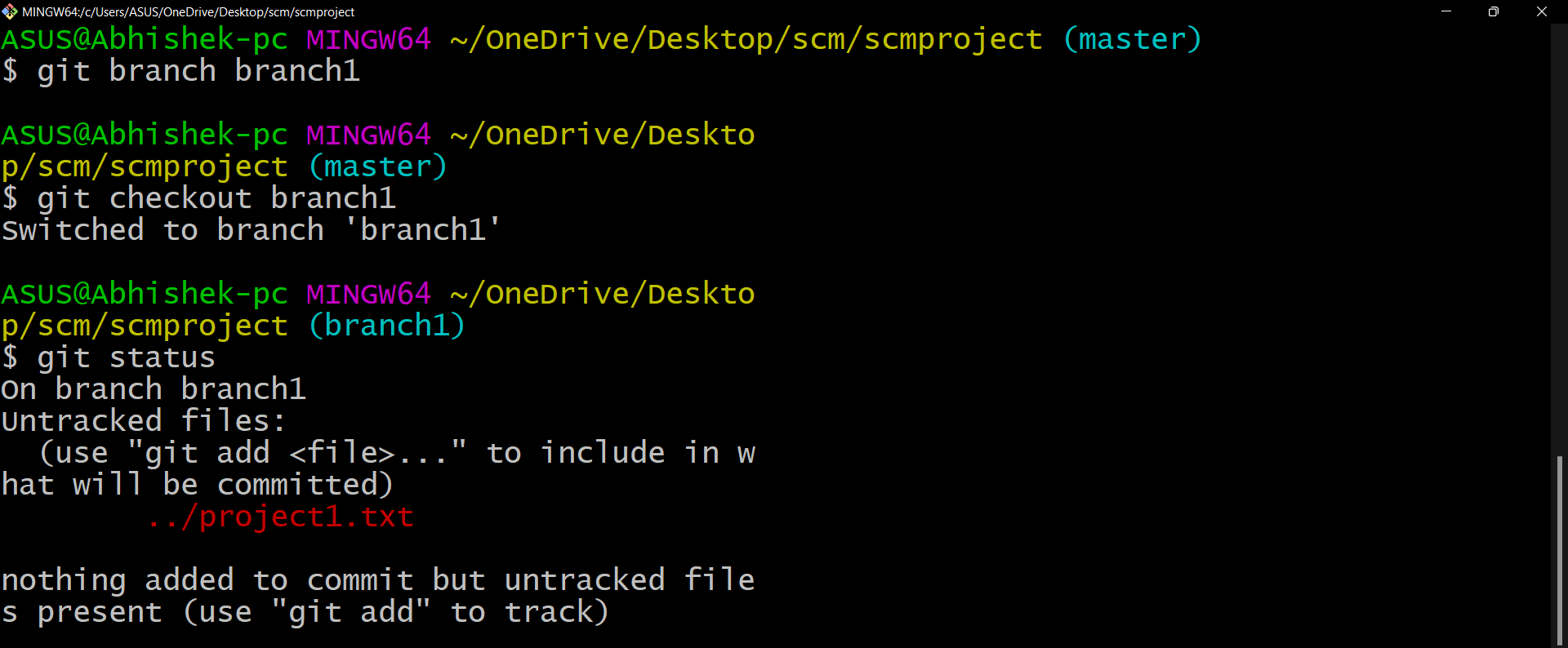


**git add: -** In order to begin tracking a new file, you use the command git add. git add is a multipurpose command — you use it to begin tracking new files, to stage files, and to do other things like marking merge-conflicted files as resolved. It may be helpful to think of it more as “add precisely this content to the next commit” rather than “add this file to the project”.

**git commit: -** The git commit command captures a snapshot of the project's currently staged changes. Committed snapshots can be thought of as “safe” versions of a project—Git will never change them unless you explicitly ask it to. Prior to the execution of git commit, the git add command is used to promote or 'stage' changes to the project that will be stored in a commit. These two commands git commit and git add are two of the most frequently used.

**git push:** - When you have your project at a point that you want to share, you have to push it upstream. The command for this is simple: git push. If you want to push your master branch to your origin server (again, cloning generally sets up both of those names for you automatically), then you can run this to push any commits you’ve done back up to the server: $ git push origin master

**Git Branching: -**

****

A branch is a version of the repository that diverges from the main working project. It is a feature available in most modern version control systems. A Git project can have more than one branch. These branches are a pointer to a snapshot of your changes.

**git branch name [**adding new branch**]**

**git branch [**use to see the branch’s names**]**

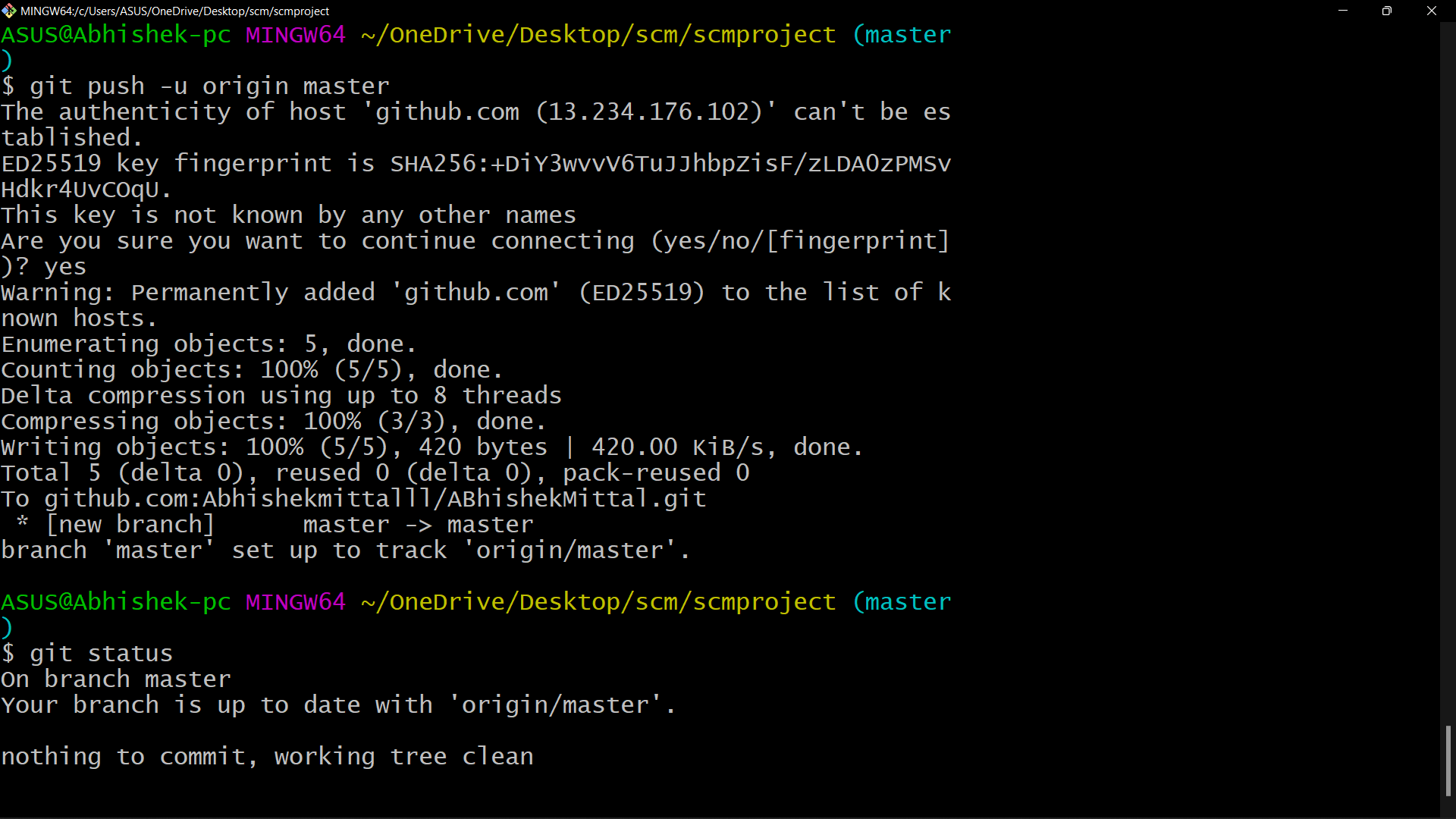
**git checkout branch name [**use to switch to the given branch]

**Pushing in git hub**

**What Does it Mean to Push to GitHub**?

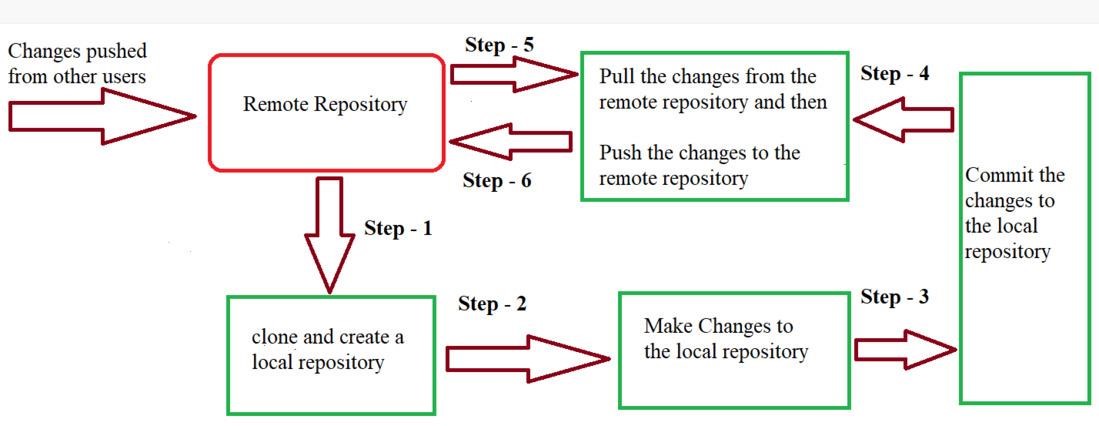
Pushing to GitHub means uploading to a GitHub repository.

Maybe you just created a new public repository for a project and you want to upload all your files to it. Or maybe you're continually working on a project from a private GitHub repository for your company, and you want to upload the latest changes that you've made.



**GIT LIFECYCLE**

It is important for us to have an abstract idea of the different stages of Git before going into more detailed understanding of Git. Files in a **Git** project have various stages like **Creation, Modification, Refactoring**, and **Deletion** and so on.



The three Git states:

* Working directory
* Staging area
* Git directory

**Working Directory:**

Consider a project residing in your local system. This project may or may not be tracked by Git. In either case, this project directory is called your Working directory.

**Staging Area:**

Staging area is the playground where you group, add and organize the files to be committed to Git for tracking their versions.

**Git Directory:**

Now that the files to be committed are grouped and ready in the staging area, we can commit these files. So, we commit this group of files along with a commit message explaining what is the commit about. Apart from commit message, this step also records the author and time of the commit. Now, a snapshot of the files in the commit is recorded by Git. The information related to this commit is stored in the Git directory.

**Remote Repository:**

Means mirror or clone of the local Git repository in GitHub. And pushing means uploading the commits from local Git repository to remote repository hosted in GitHub