Getting Started with GitHub

Chameleon Security Team project

Table of Content

1.0 INTRODUCTION TO GITHUB

What is GitHub:

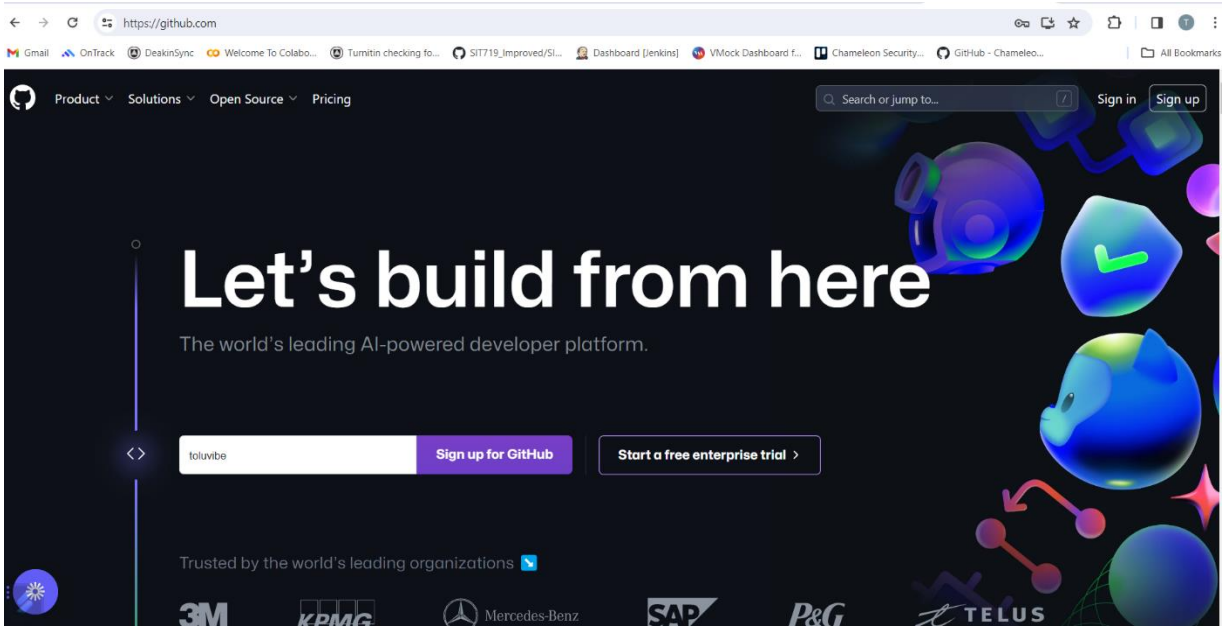
GitHub is a web-based platform that hosts software development projects and uses Git for version management. GitHub offers a user-friendly interface, which is a very collaborative tool, and more project management tools that allow developers to create and manage the code in the repository in a remote location where others can access the code.

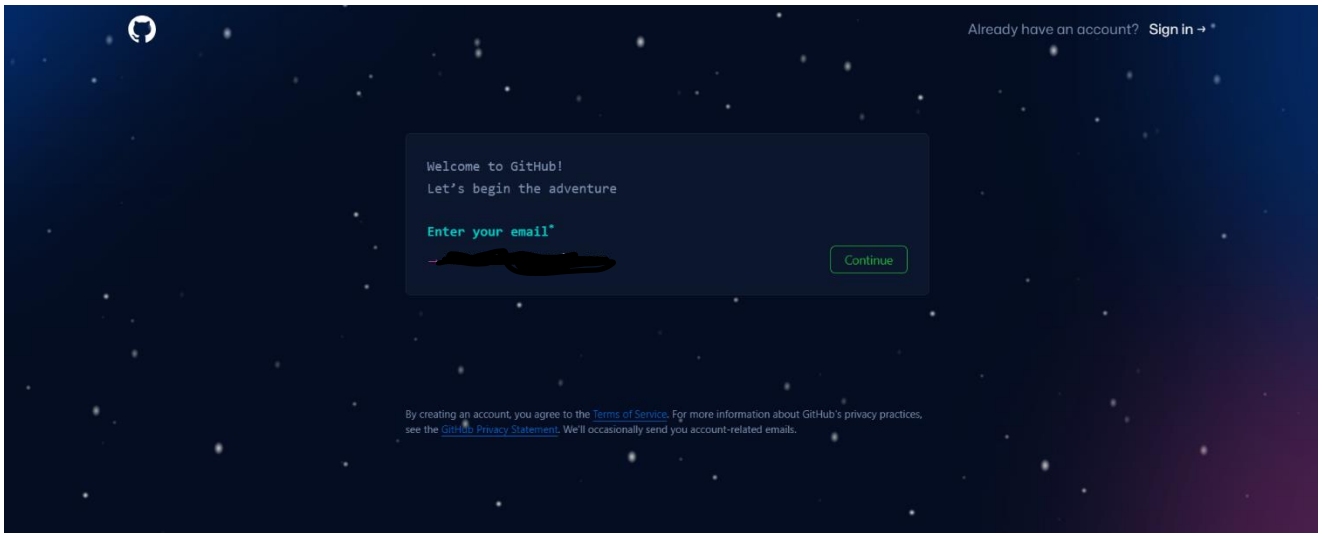
What is Git:

Git is a free and open-source distributed version control system that helps developers work together on the same software projects and keep track of changes made to their code by another.

2.0 SETTING UP A GITHUB ACCOUNT

- Go to https://github.com and enter the required user credentials asked on the site and then click on the SignUp for GitHub button.

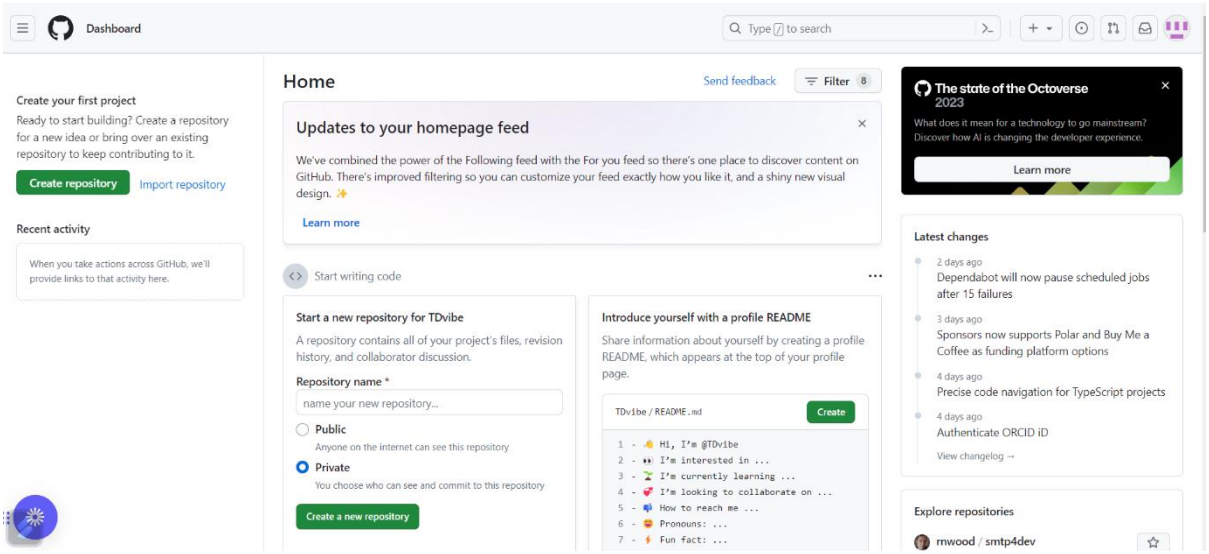




Then enter your email and create a password.

A verification link will be sent to your email, check your email, and verify the link.

You will be redirected to the GitHub dashboard, take your time to explore the dashboard.



3.0 SETTING UP GIT

Steps to install Git on Windows and MacOS:

- Download the executable Git file from https://git-scm.com/downloads.
- Run the installation file with Administrator rights.
- Choose an appropriate installation location such as C:\SIT764\git.
- Install the default components, including Git GUI Here and Git Bash Here.
- Choose your preferred Git default editor.
- Add Git to the Windows PATH.
- Accept the default line ending conversion for Unix and Windows compatibility.
- Choose the extra option to enable system caching.
- Click Finish to complete the install.
- To ensure that Git has been installed correctly, open Git Bash and type: git --version
- Open a Git Bash shell and start using Git

Understanding common Git commands:

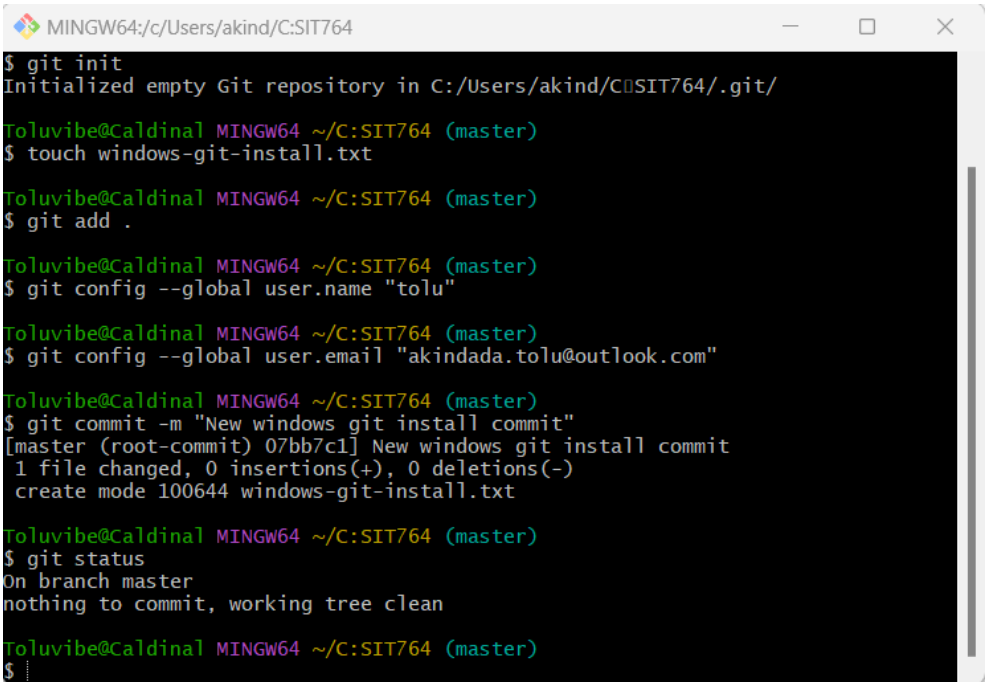
- **init**
- **add**
- **config**
- **status**
- **commit**

Before you run Git init, first create a new folder named SIT764 off the C:\ drive using Git Bash: **mkdir C:\SIT764**

After creating the folder, you can navigate into it using the cd command: **cd C:\SIT764**

1. Initialize a new Git repository if needed using **git init**
2. In the newly created folder, add a file named “windows-git-install.txt” using **touch windows-git-install.txt**
3. Then add the file to the Git index using **git add .**
4. Configure the Git user’s name: **git config --global user.name "type name"**
5. Configure the Git user’s email address: **git config --global user.email "type email"**
6. Commit the new file to the new Git repository: **git commit -m "New Windows git install commit"**

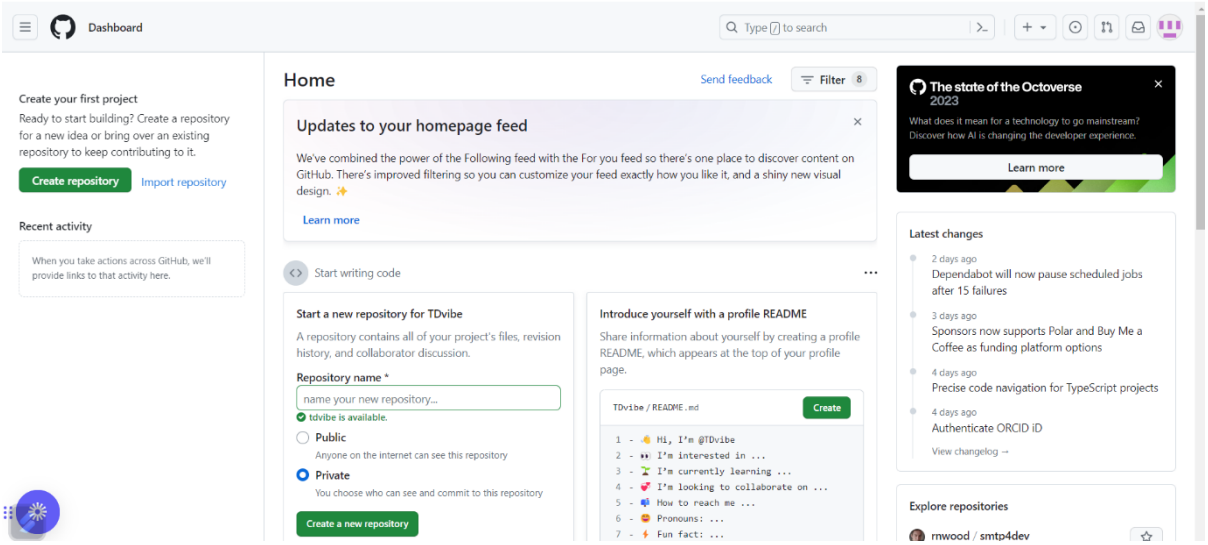
7. check the status of the newly installed Git repo: **git status**



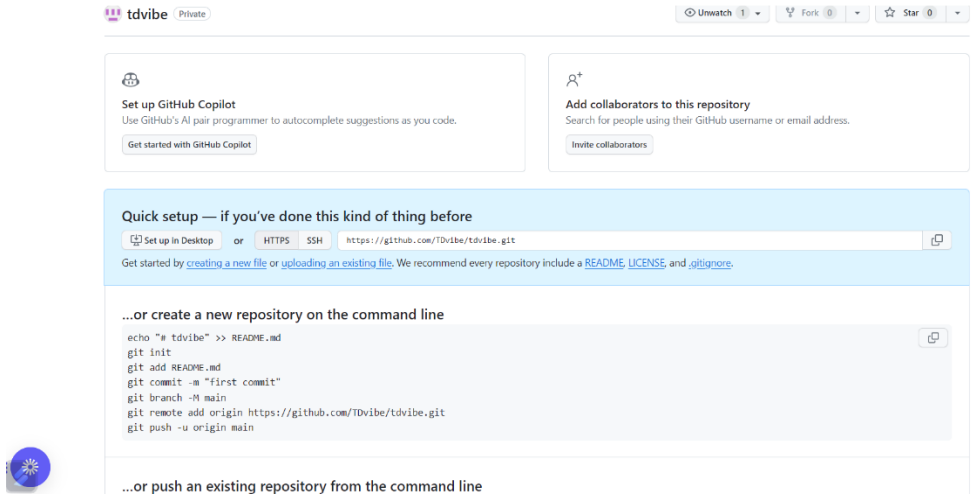
Git is completed set up and we can now continue using the basic command.

4.0 CREATING AND MANAGING REPOSITORIES

- Login to your GitHub account.
- starting a new repository.
- Give a repository name and select the visibility and accessibility mode for the repository
- Click on Create repository



New repository dashboard would be like this below:



5.0 LINKING GIT TO THE CHAMELEON SECURITY TEAM REPOSITORY

Git can share code with others. However, to work on an existing repository that Chameleon Security hosted on GitHub, you simply obtain the unique address of the repository and perform a clone.

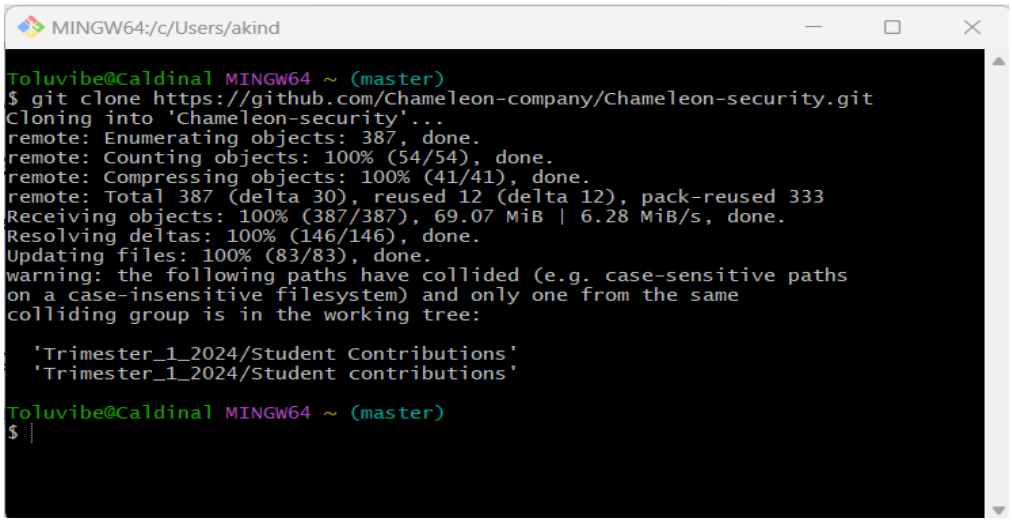
The chameleon security repository is given below:

git clone https://github.com/Chameleon-company/Chameleon-security.git

Steps to follow:

- Open git bash shell
- Use clone command: git clone https://github.com/Chameleon-company/Chameleon-security.git

The following image will appear once Chameleon Security GitHub repository has been linked to git on your computer.



Uploading files to GitHub repository can be done by clicking “add files” on github and uploading project files.

6.0 TERMINOLOGY

- pull: Pull refers to when you are fetching in changes and merging them
- merge: Merging takes the changes from one branch (in the same repository or from a fork) and applies them into another.
- fork: A fork is a personal copy of another user's repository that lives on your account
- repository: They're easiest to imagine as a project's folder. A repository contains all of the project files (including documentation) and stores each file's revision history. Repositories can have multiple collaborators and can be either public or private and so much more.
  More of these terminologies can be found in the link given below:
  https://docs.github.com/en/get-started/learning-about-github/github-glossary

THANK YOU FOR READING AND I HOPE YOU FOUND IT HELPFUL IN YOUR PROJECT WITH THE COMPANY.

REFERENCES

- https://www.geeksforgeeks.org/introduction-to-github
- https://www.theserverside.com/blog/Coffee-Talk-Java-News-Stories-and-Opinions/Step-by-step-guide-to-install-Git-on-Windows-desktop-computers
- https://digital.gov/resources/an-introduction-github/
- https://docs.github.com/en/get-started/learning-about-github/github-glossary