# **DEVOPS EXPERIMENT 3**

**AIM:** To perform various GIT operations on local and Remote repositories using GIT cheat sheet

**LO:** L2-To obtain complete knowledge of the “version control system” to effectively track changes augmented with Git and GitHub

**Theory:**

**Git**  is software for tracking changes in any set of [files](https://en.wikipedia.org/wiki/Computer_file), usually used for coordinating work among [programmers](https://en.wikipedia.org/wiki/Programmer) collaboratively developing [source code](https://en.wikipedia.org/wiki/Source_code) during [software development](https://en.wikipedia.org/wiki/Software_development). Its goals include speed, [data integrity](https://en.wikipedia.org/wiki/Data_integrity), and support for distributed, non-linear workflows (thousands of parallel branches running on different systems).Git was originally authored by [Linus Torvalds](https://en.wikipedia.org/wiki/Linus_Torvalds) in 2005 for development of the [Linux kernel](https://en.wikipedia.org/wiki/Linux_kernel), with other kernel developers contributing to its initial development.Since 2005, Junio Hamano has been the core maintainer. As with most other [distributed version control](https://en.wikipedia.org/wiki/Distributed_version_control) systems, and unlike most [client–server](https://en.wikipedia.org/wiki/Client%E2%80%93server) systems, every Git [directory](https://en.wikipedia.org/wiki/Directory_(computing)) on every [computer](https://en.wikipedia.org/wiki/Node_(networking)) is a full-fledged [repository](https://en.wikipedia.org/wiki/Repository_(version_control)) with complete history and full version-tracking abilities, independent of network access or a central server.Git is [free and open-source software](https://en.wikipedia.org/wiki/Free_and_open-source_software) distributed under the [GPL-2.0-only](https://en.wikipedia.org/wiki/GNU_General_Public_License) license.

Git also has excellent support for branching, merging, and rewriting repository history, which has led to many innovative and powerful workflows and tools. Pull requests are one such popular tool that allows teams to collaborate on Git branches and efficiently review each other's code. Git is the most widely used version control system in the world today and is considered the modern standard for software development.

**Features of Git**

* Tracks history
* Free and open source
* Supports non-linear development
* Creates backups
* Scalable
* Supports collaboration
* Branching is easier
* Distributed development

#### Create your local Git repository

In your computer, create a folder for your project. Let’s call the project folder simple-git-demo. Go into your project folder and add a local Git repository to the project using the following command - git init

**Add some demo file**

Create a file called demo.txt in the project folder and add some textin it

### **Staging and Committing the code**

Committing is the process in which the code is added to the **local repository**. Before committing the code, it has to be in the **staging area**. The staging area is there to keep track of all the files which are to be committed. Any file which is not added to the staging area will not be committed. This gives the developer control over which files need to be committed.

#### Staging

Use the following command for staging the file: git add demo.txt

**Committing**

Use the following command to commit the file: git commit -m "Initial Commit"

“Initial Commit” is the commit message here. Enter a relevant commit message to indicate what code changes were done in that particular commit.

### **Git Status and Git Log**

Now modify the demo.txtfile and add more text like : Initial Content Adding more Content

#### Status

Use git status to find out information regarding what files are modified and what files are there in the staging area — it shows other information as well, which we can ignore for now.

Use the following command to see the status: git status

The status shows that demo.txt is modified and is not yet in the staging area.

Now let us add demo.txt to the staging area and commit it using the following commands:git add demo.txt git commit -m "demo.txt file is modified"

**Log**

Use git log to print out all the commits which have been done up until now.

The command used for this is: git log

The log shows the author of each commit, the date of the commit, and the commit message.

**Github**

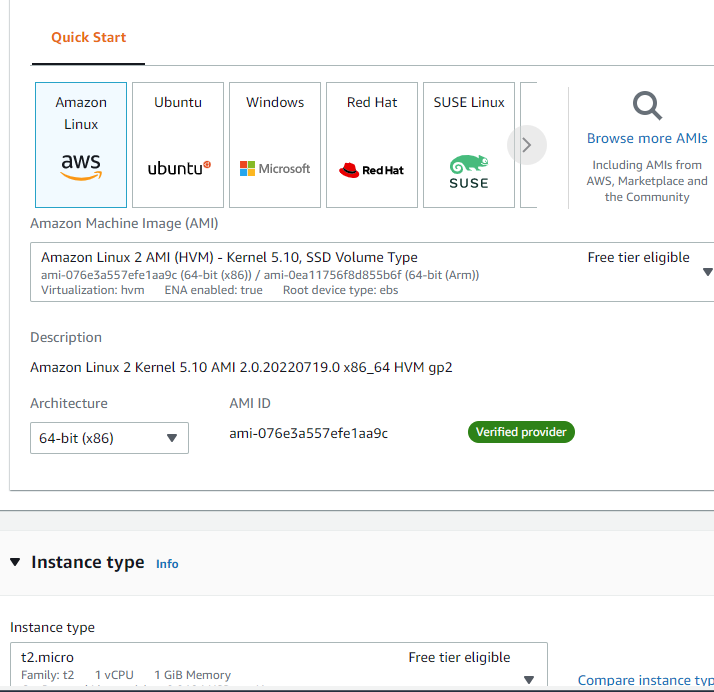
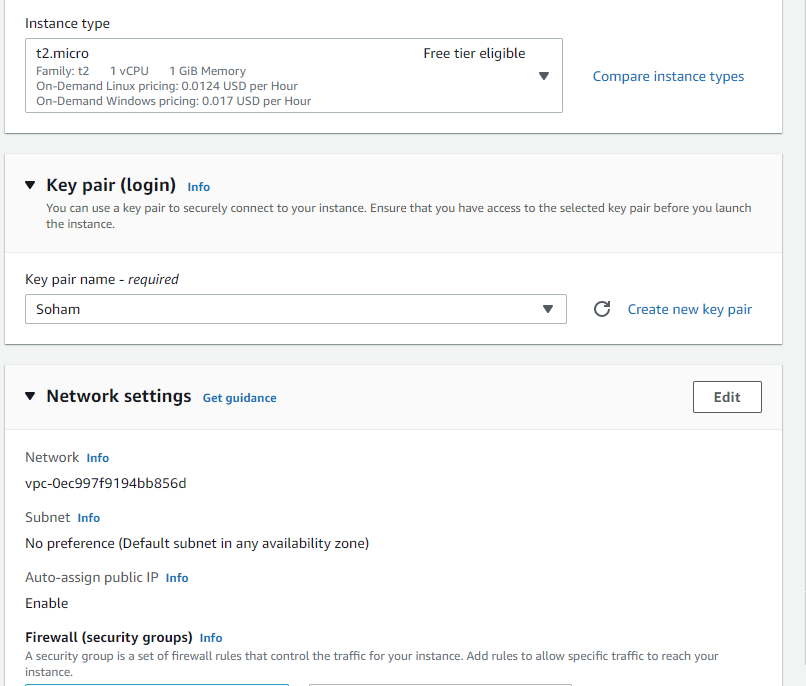
[GitHub](https://github.com/) is a web-based interface that uses [Git](https://git-scm.com/), the open source version control software that lets multiple people make separate changes to web pages at the same time. As Carpenter notes, because it allows for real-time collaboration, GitHub encourages teams to work together to build and edit their site content.

Git is used to [store the source code for a project](https://www.techtarget.com/searchvmware/tip/How-to-use-GitHub-vRA-integration-to-keep-track-of-blueprints) and track the complete history of all changes to that code. It allows developers to collaborate on a project more effectively by providing tools for managing possibly conflicting changes from multiple developers. GitHub allows developers to change, adapt and improve software from its public repositories for free, but it charges for private repositories, offering various paid plans. Each public or private repository contains all of a project's files, as well as each file's revision history. Repositories can have multiple collaborators and can be either public or private.

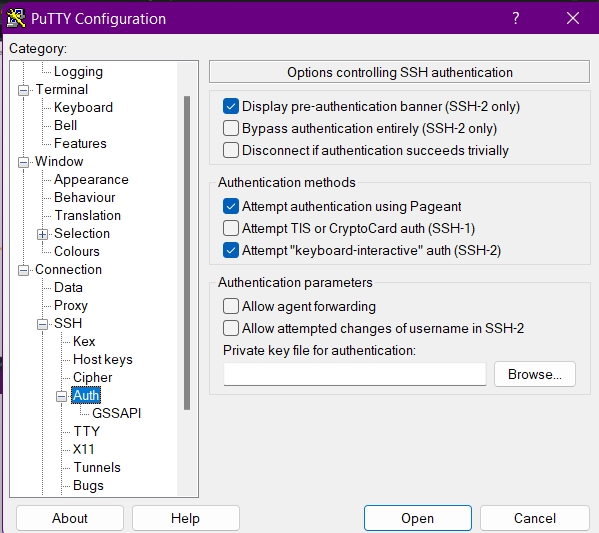
GitHub facilitates [social coding](https://www.techtarget.com/whatis/definition/social-coding) by providing a web interface to the Git [code](https://www.techtarget.com/whatis/definition/code) [repository](https://www.techtarget.com/searchoracle/definition/repository) and management tools for collaboration. GitHub can be thought of as a serious [social networking](https://www.techtarget.com/whatis/definition/social-networking) site for software developers. Members can follow each other, rate each other's work, receive updates for specific projects and communicate publicly or privately.

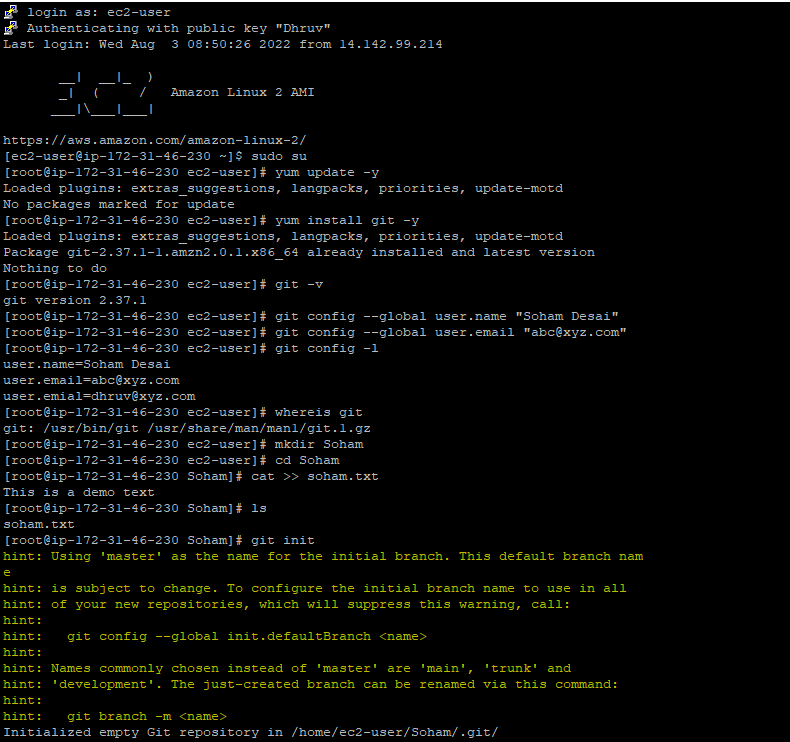
**Output:**

1. Creating instance in AWS EC2

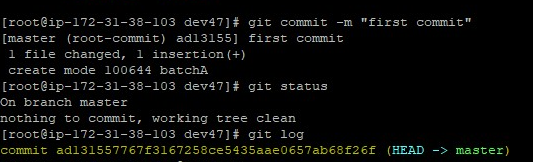


1. Logging into the instance with PuTTy

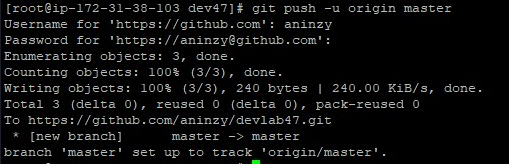




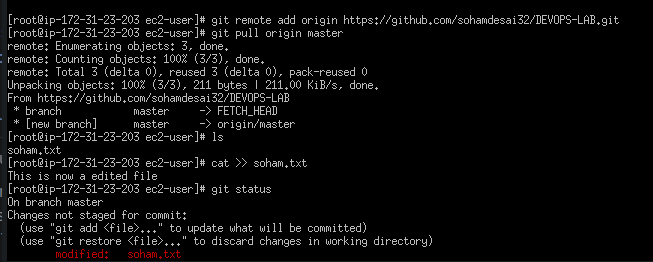
1. Creating a file and running git commands



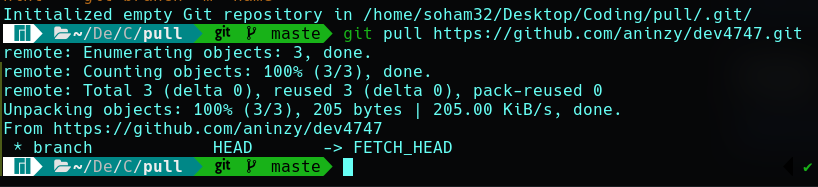
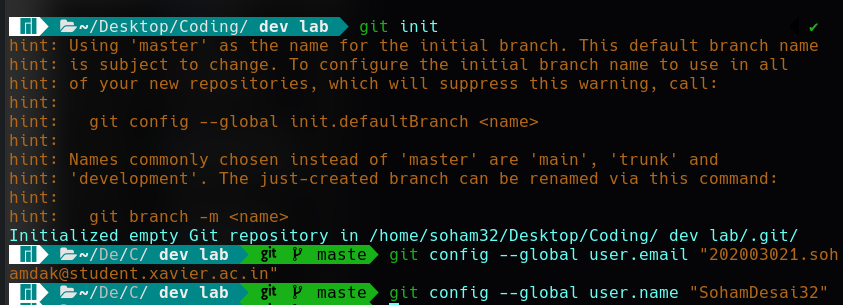
1. Pushing the files on GitHub

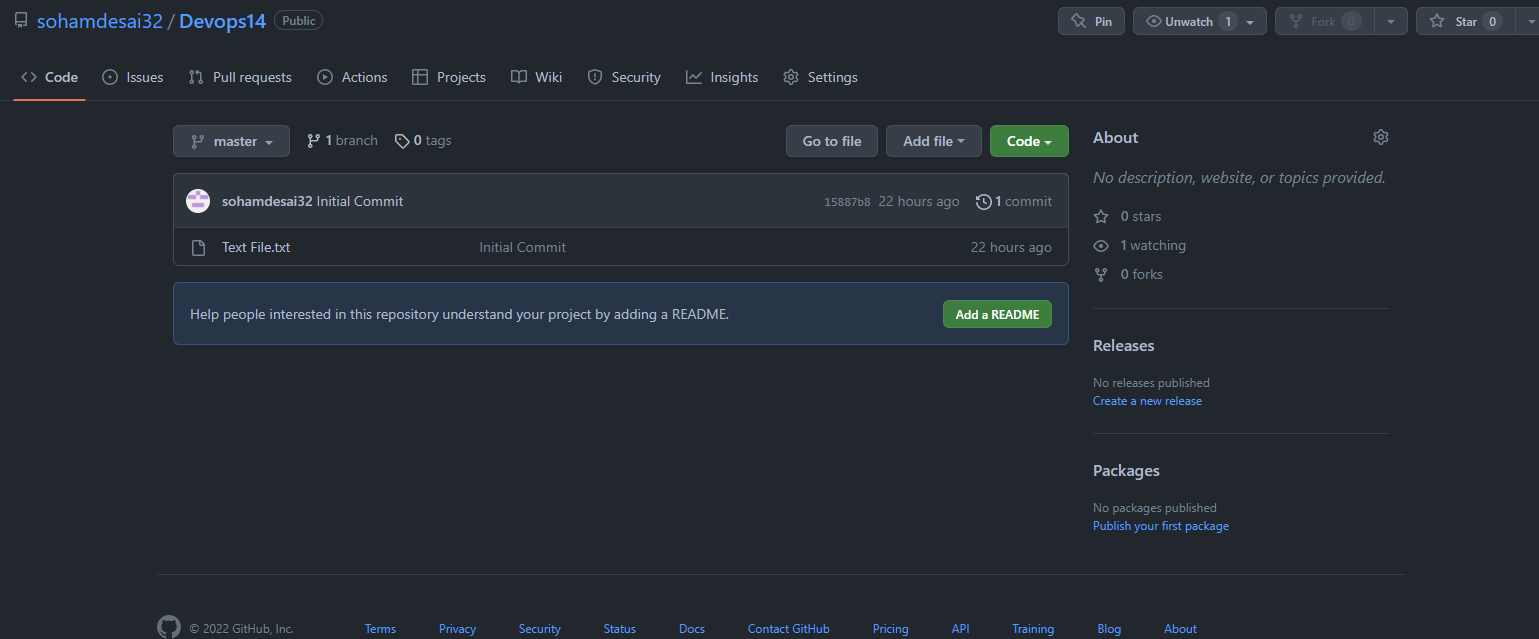


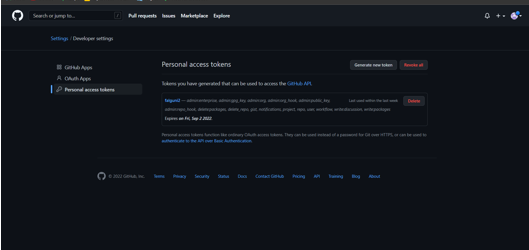
1. Pulling the files to make changes and again pushing

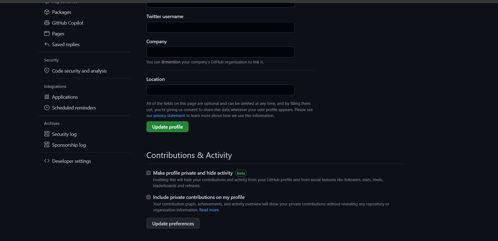


1. Pulling and pushing the files from a repository using git on the device



1. Repository on github
2. Token generation





**Conclusion:** From this experiment we have learned several GIT commands and also how to use the GitHub to host our files