MCQ

What command is used to initialize a Git repository locally?

a) git clone

**b) git init**

c) git commit

d) git push

How can you check the status of your changes in a Git repository?

**a) git status**

b) git check

c) git diff

d) git log

What command is used to stage files for a commit in Git?

**a) git add**

b) git stage

c) git commit

d) git push

What is the purpose of forking a repository on GitHub?

**a) To create a new branch in the original repository**

b) To merge changes from one repository to another

c) To copy a repository under your GitHub account

d) To revert changes in a repository

What is a Pull Request used for in GitHub?

**a) Requesting changes to be pulled into a repository**

b) Submitting changes for approval and merging

c) Deleting branches in a repository

d) Checking the status of commits in a repository

What does a 'Merge Conflict' indicate in a GitHub pull request?

a) Successful merging of changes

**b) Inconsistencies between branches that need to be resolved**

c) Rejection of a pull request

d) Approval of changes for merging

Which command is used to create a new branch in Git?

**a) git branch**

b) git commit

c) git checkout

d) git merge

What command is used to view the commit history in Git?

**a) git log**

b) git history

c) git show

d) git status

How can you undo the last commit in Git?

a) git amend

**b) git reset**

c) git revert

d) git undo

What is a repository in GitHub?

a) A folder on your local machine

**b) A collection of project files and revision history**

c) A social media platform for developers

d) A code editor tool

How can you clone a repository from GitHub to your local machine?

**a) git clone**

b) git fetch

c) git init

d) git pull

What is the purpose of the 'Issues' tab in GitHub repositories?

**a) To track and discuss bugs, enhancements, and tasks**

b) To view commit history

c) To create new branches

d) To merge changes into the main branch

Which GitHub feature allows multiple people to collaborate on a project simultaneously?

a) Pull Requests

b) Forking

c) Issues

**d) Branches**

What does the 'README.md' file in a GitHub repository contain?

**a) Detailed instructions for using the project**

b) A list of contributors

c) Commit history

d) License information

How can you update your local repository with changes from a remote repository in Git?

a) git merge

**b) git fetch**

c) git update

d) git commit -u origin

What is the purpose of the 'git push' command in Git?

a) To stage changes for commit

b) To download changes from a remote repository

**c) To update the remote repository with local changes**

d) To switch between branches

What is 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.

What is CVCS and DVCS ?

**Centralized Version Control System(CVCS):**

In centralized source control, there is a server and a client. The server is the master repository that contains all of the versions of the code. To work on any project, firstly user or client needs to get the code from the master repository or server. So the client communicates with the server and pulls all the code or current version of the code from the server to their local machine. In other terms we can say, you need to take an update from the master repository and then you get the local copy of the code in your system. So once you get the latest version of the code, you start making your own changes in the code and after that, you simply need to commit those changes straight forward into the master repository. Committing a change simply means merging your own code into the master repository or making a new version of the source code. So everything is centralized in this model.   
There will be just one repository and that will contain all the history or version of the code and different branches of the code. So the basic workflow involves in the centralized source control is getting the latest version of the code from a central repository that will contain other people’s code as well, making your own changes in the code, and then committing or merging those changes into the central repository.

**Distributed Version Control System(DVCS):**

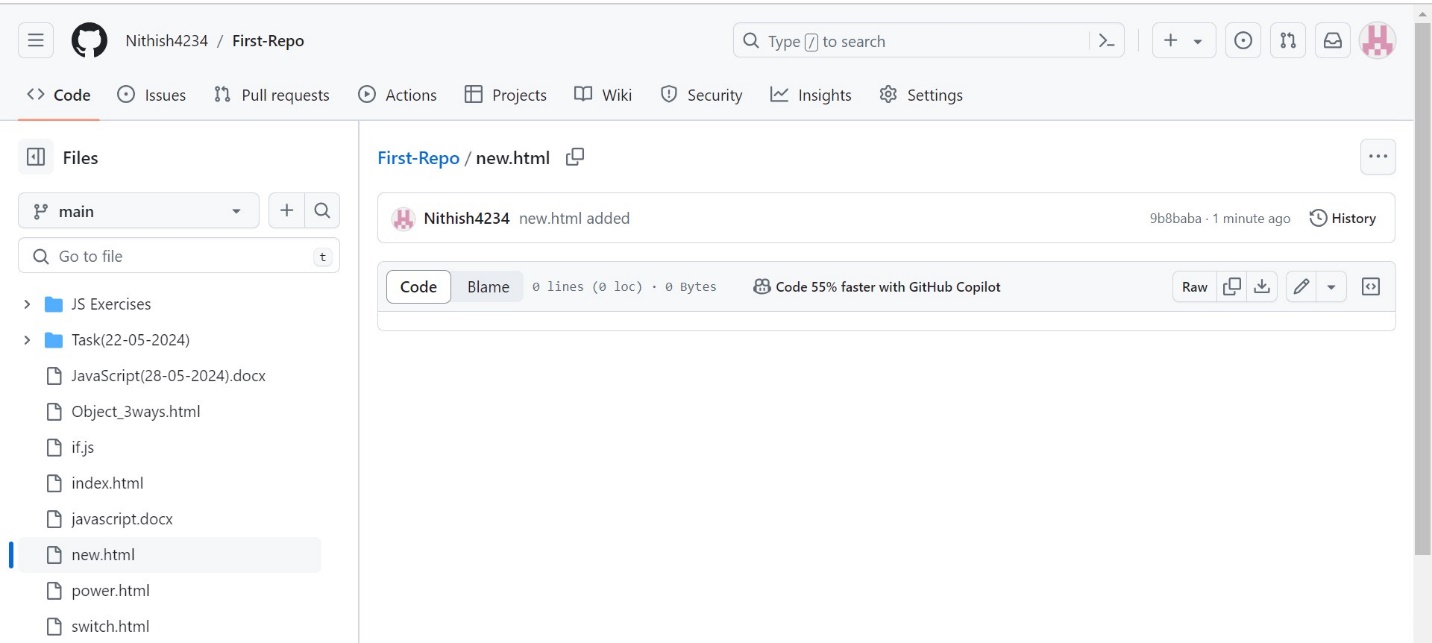
In distributed version control most of the mechanism or model applies the same as centralized. The only major difference you will find here is, instead of one single repository which is the server, here every single developer or client has their own server and they will have a copy of the entire history or version of the code and all of its branches in their local server or machine. Basically, every client or user can work locally and disconnected which is more convenient than centralized source control and that’s why it is called distributed.   
You don’t need to rely on the central server, you can clone the entire history or copy of the code to your hard drive. So when you start working on a project, you clone the code from the master repository in your own hard drive, then you get the code from your own repository to make changes and after doing changes, you commit your changes to your local repository and at this point, your local repository will have ‘*change sets*‘ but it is still disconnected with the master repository (master repository will have different ‘**sets of changes**‘ from each and every individual developer’s repository), so to communicate with it, you issue a request to the master repository and push your local repository code to the master repository. Getting the new change from a repository is called “**pulling**” and merging your local repository’s ‘set of changes’ is called “**pushing**“.   
It doesn’t follow the way of communicating or merging the code straight forward to the master repository after making changes. Firstly you commit all the changes in your own server or repository and then the ‘set of changes’ will merge to the master repository.

Create a project of any and push the project

Commands to push:

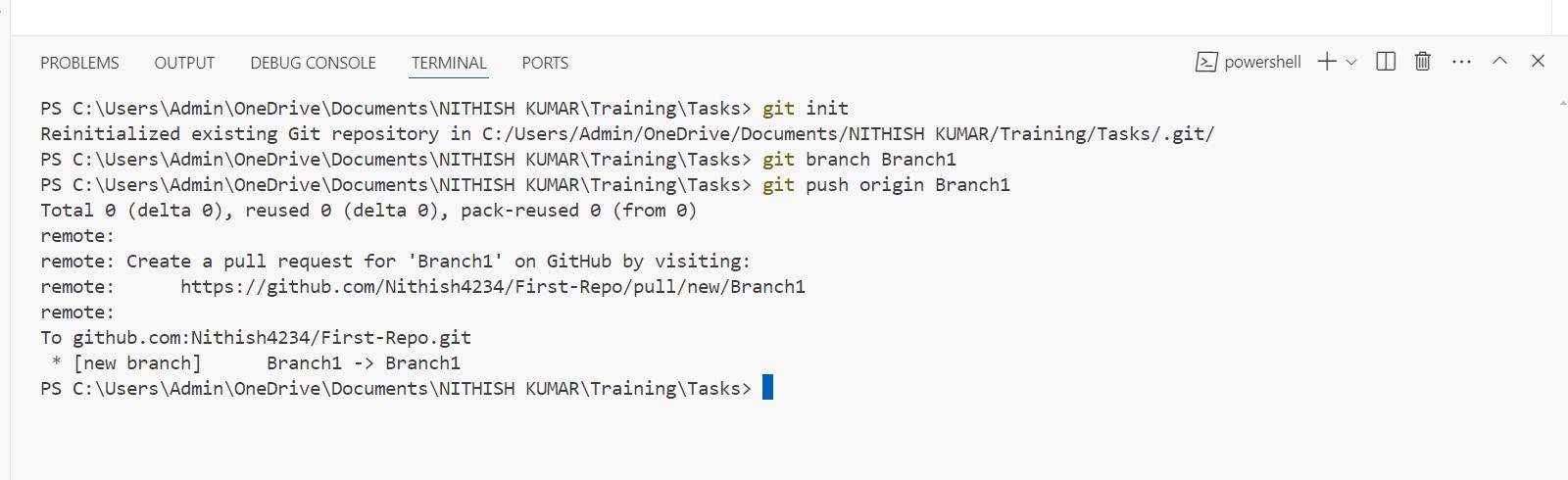


After pushing to repository:

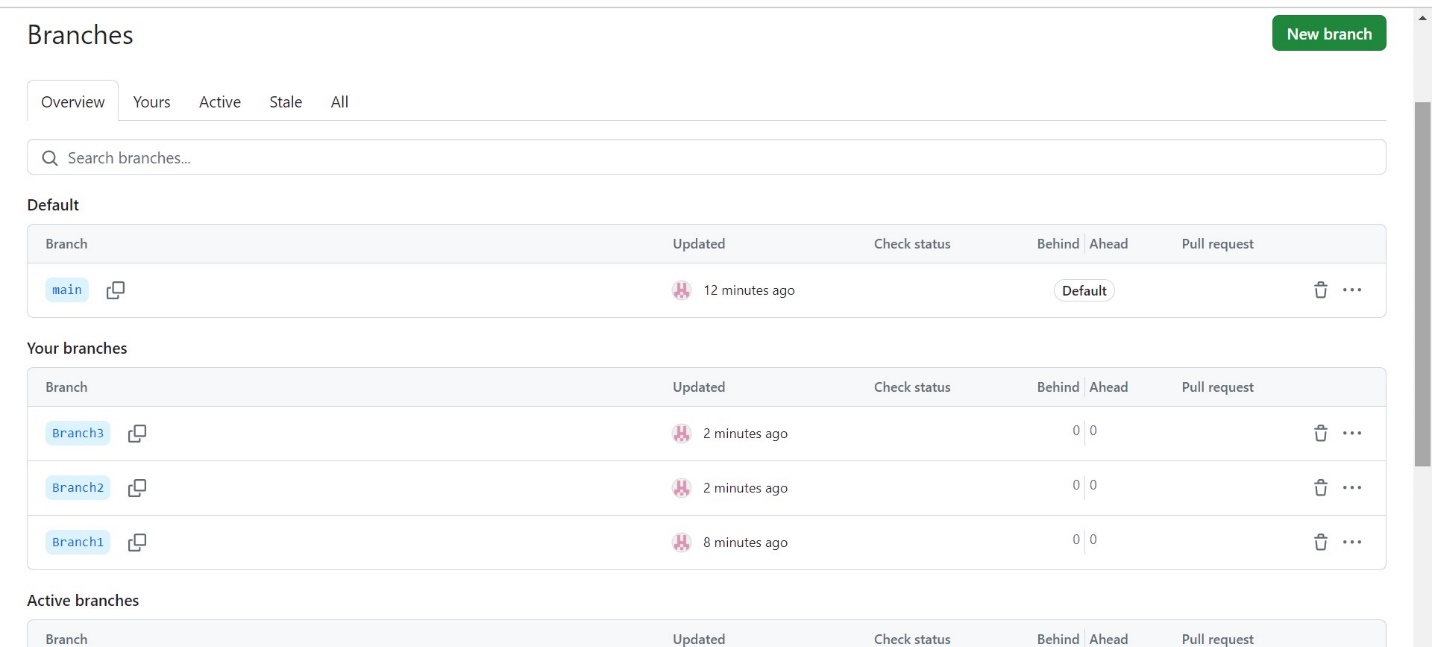


Create 3 branches and 5 tags

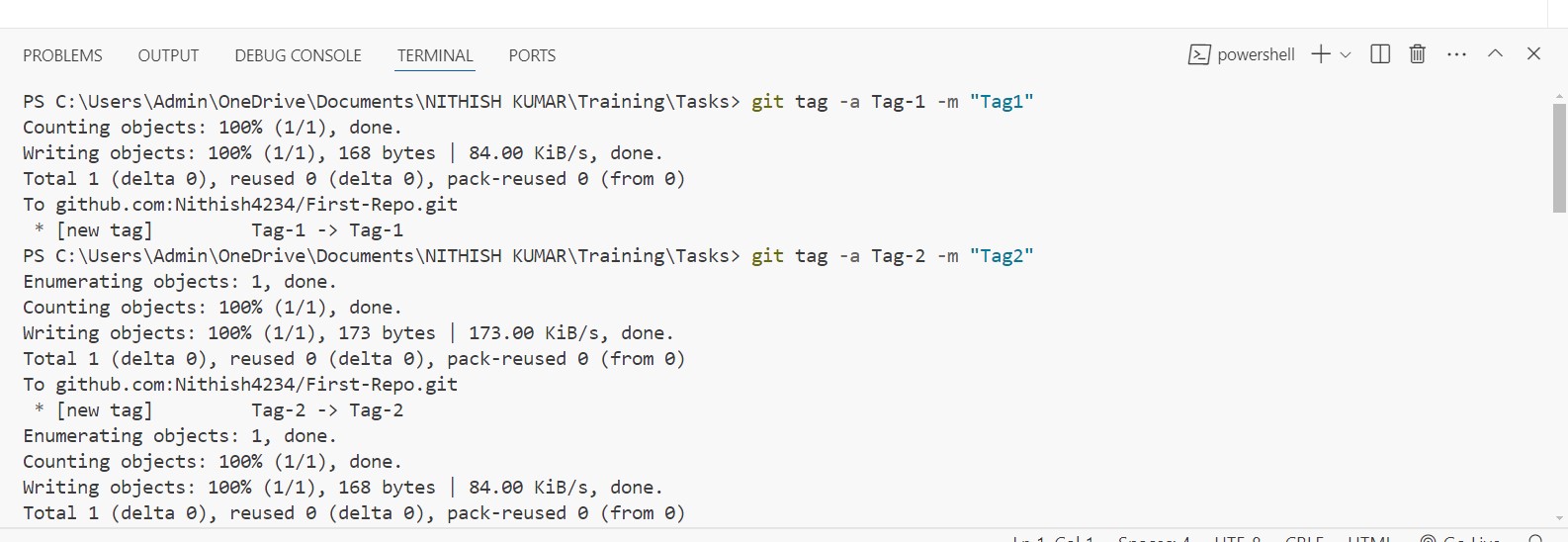
Commands for creating branch:



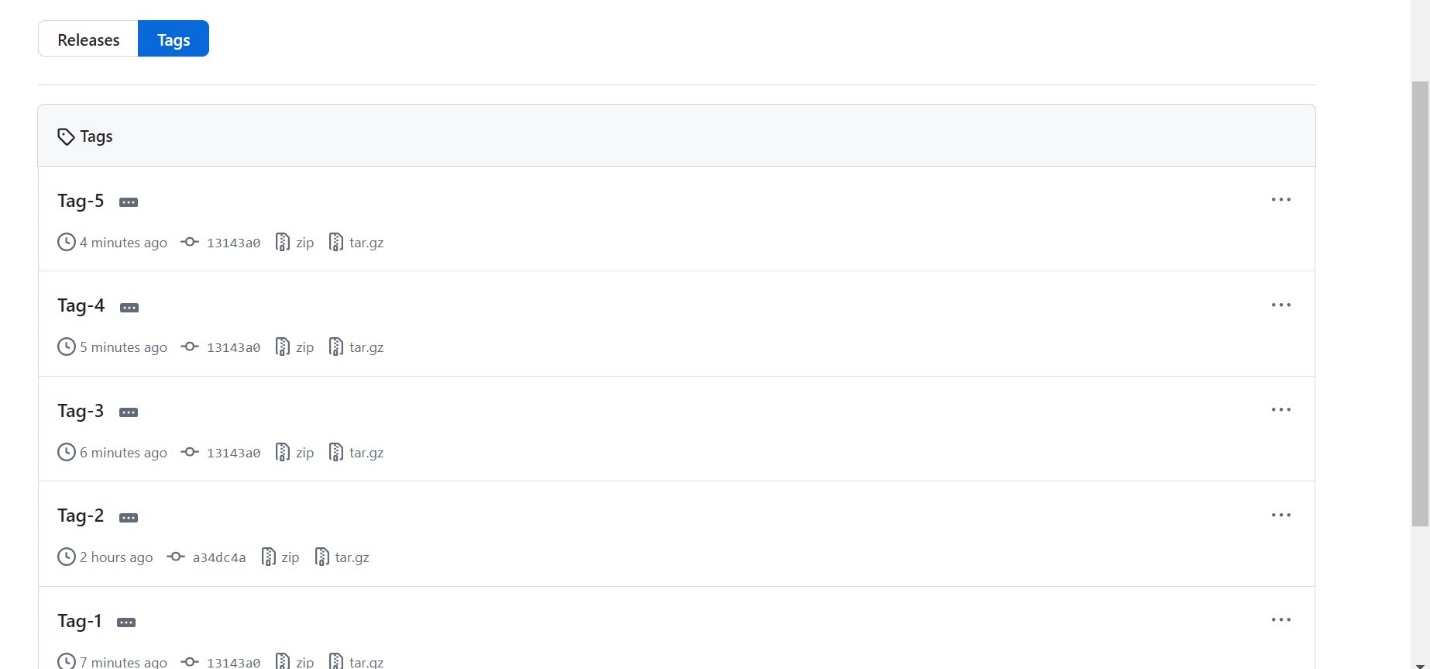
Created branches:



Commands for creating Tags:

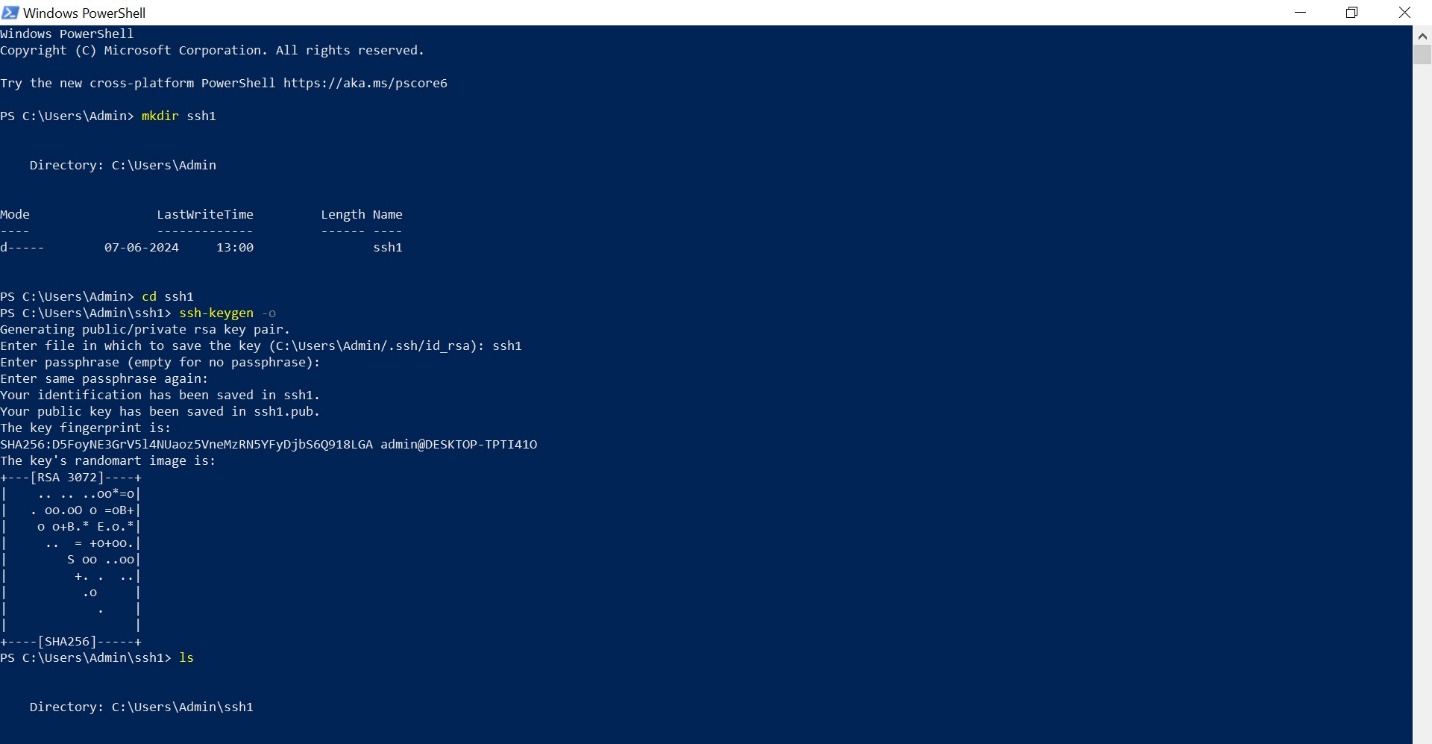


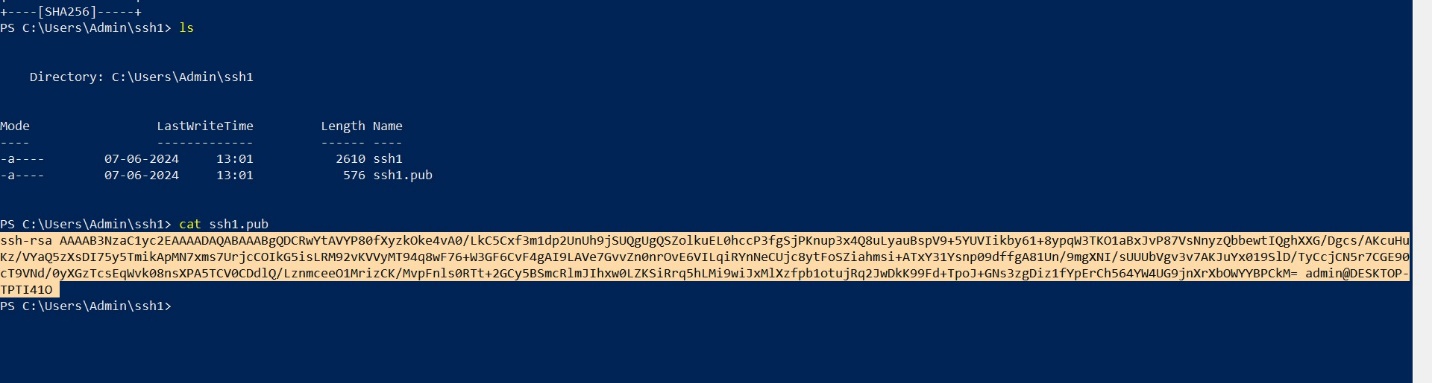
Created Tags:



Create a Keygen and push using ssh

Commands for generating and adding key:





After adding key:

