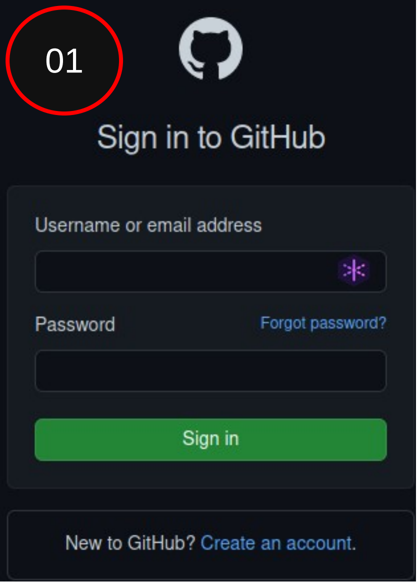
Module-2: GIT Assignment - 1

Based on what you have learnt in the class, do the following steps:

- Create a new folder
- Put the following files in the folder
  - Code.txt
  - Log.txt
  - Output.txt
- Stage the Code.txt and Output.txt files
- Commit them
- And finally push them to github

Please share the commands for the above points



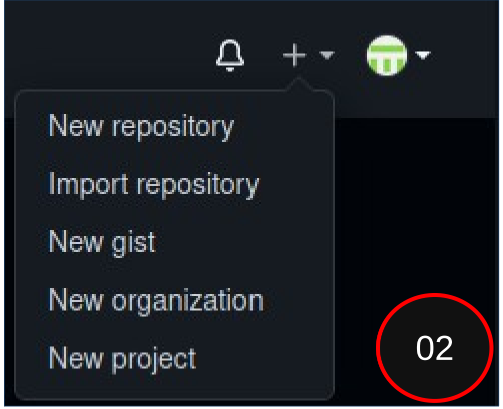
Sign in to GitHub

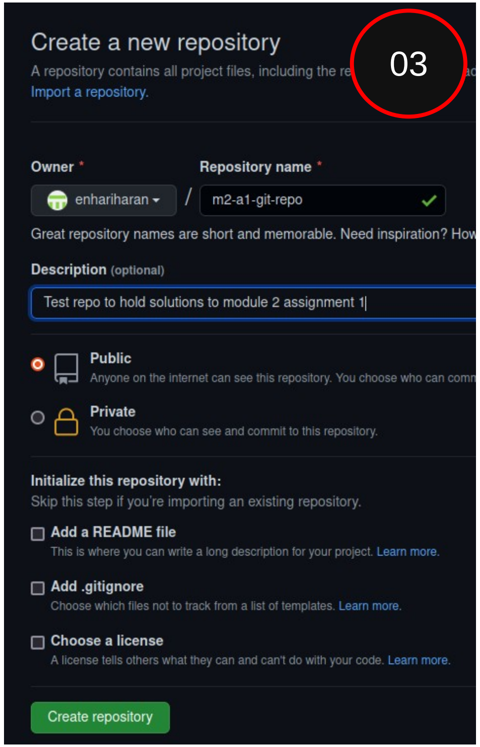
Username or email address

Password Forgot password?

Sign in

New to GitHub? Create an account.





Create a new repository

A repository contains all project files, including the re
Import a repository.

Owner * Repository name *

enhariharan / m2-a1-git-repo ✓

Great repository names are short and memorable. Need inspiration? How

Description (optional)

Test repo to hold solutions to module 2 assignment 1|

● Public
Anyone on the internet can see this repository. You choose who can comm

○ Private
You choose who can see and commit to this repository.

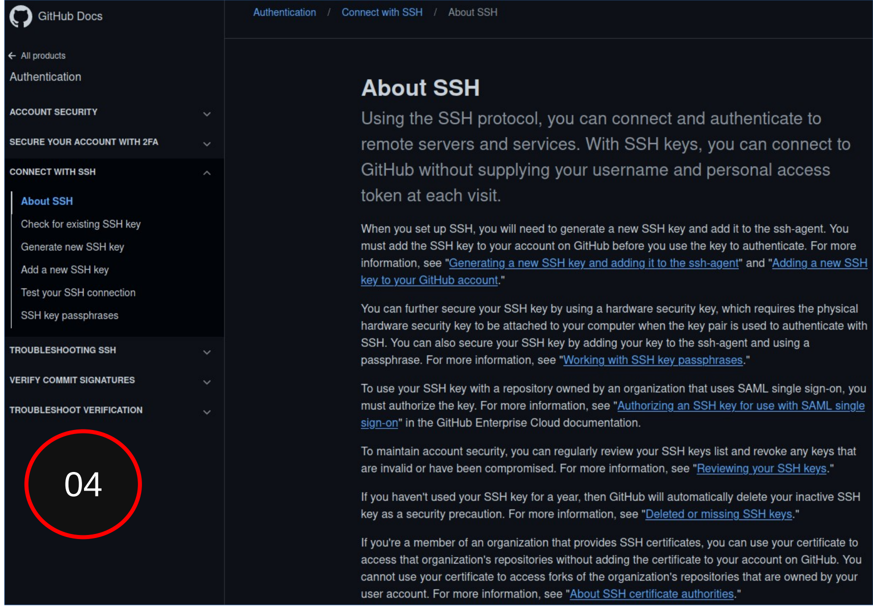
Initialize this repository with:
Skip this step if you're importing an existing repository.

□ Add a README file
This is where you can write a long description for your project. Learn more.

□ Add .gitignore
Choose which files not to track from a list of templates. Learn more.

□ Choose a license
A license tells others what they can and can't do with your code. Learn more.

Create repository



GitHub Docs

Authentication / Connect with SSH / About SSH

← All products
Authentication

ACCOUNT SECURITY
SECURE YOUR ACCOUNT WITH 2FA
CONNECT WITH SSH
- About SSH
- Check for existing SSH key
- Generate new SSH key
- Add a new SSH key
- Test your SSH connection
- SSH key passphrases

TROUBLESHOOTING SSH
VERIFY COMMIT SIGNATURES
TROUBLESHOOT VERIFICATION

About SSH

Using the SSH protocol, you can connect and authenticate to remote servers and services. With SSH keys, you can connect to GitHub without supplying your username and personal access token at each visit.

When you set up SSH, you will need to generate a new SSH key and add it to the ssh-agent. You must add the SSH key to your account on GitHub before you use the key to authenticate. For more information, see "Generating a new SSH key and adding it to the ssh-agent" and "Adding a new SSH key to your GitHub account."

You can further secure your SSH key by using a hardware security key, which requires the physical hardware security key to be attached to your computer when the key pair is used to authenticate with SSH. You can also secure your SSH key by adding your key to the ssh-agent and using a passphrase. For more information, see "Working with SSH key passphrases."

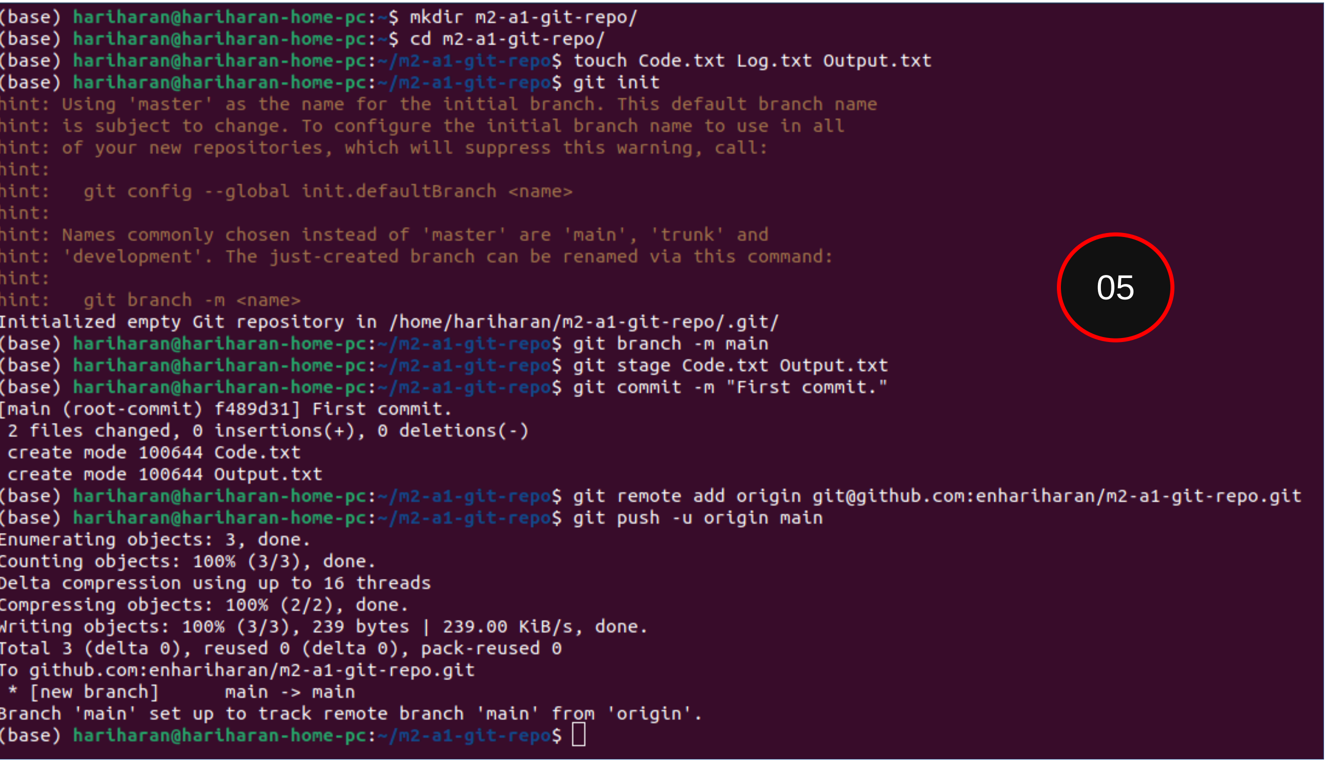
To use your SSH key with a repository owned by an organization that uses SAML single sign-on, you must authorize the key. For more information, see "Authorizing an SSH key for use with SAML single sign-on" in the GitHub Enterprise Cloud documentation.

To maintain account security, you can regularly review your SSH keys list and revoke any keys that are invalid or have been compromised. For more information, see "Reviewing your SSH keys."

If you haven't used your SSH key for a year, then GitHub will automatically delete your inactive SSH key as a security precaution. For more information, see "Deleted or missing SSH keys."

If you're a member of an organization that provides SSH certificates, you can use your certificate to access that organization's repositories without adding the certificate to your account on GitHub. You cannot use your certificate to access forks of the organization's repositories that are owned by your user account. For more information, see "About SSH certificate authorities."

1. Go to https://github.com/login and login into GitHub
2. Select “New Repository”
3. Name the new repo “m2-a1-git-repo”. Keep it Public. Click on “Create repository”.
4. Go to https://docs.github.com/en/authentication/connecting-to-github-with-ssh/about-ssh and create SSH keys to do SSH logins into GitHub.

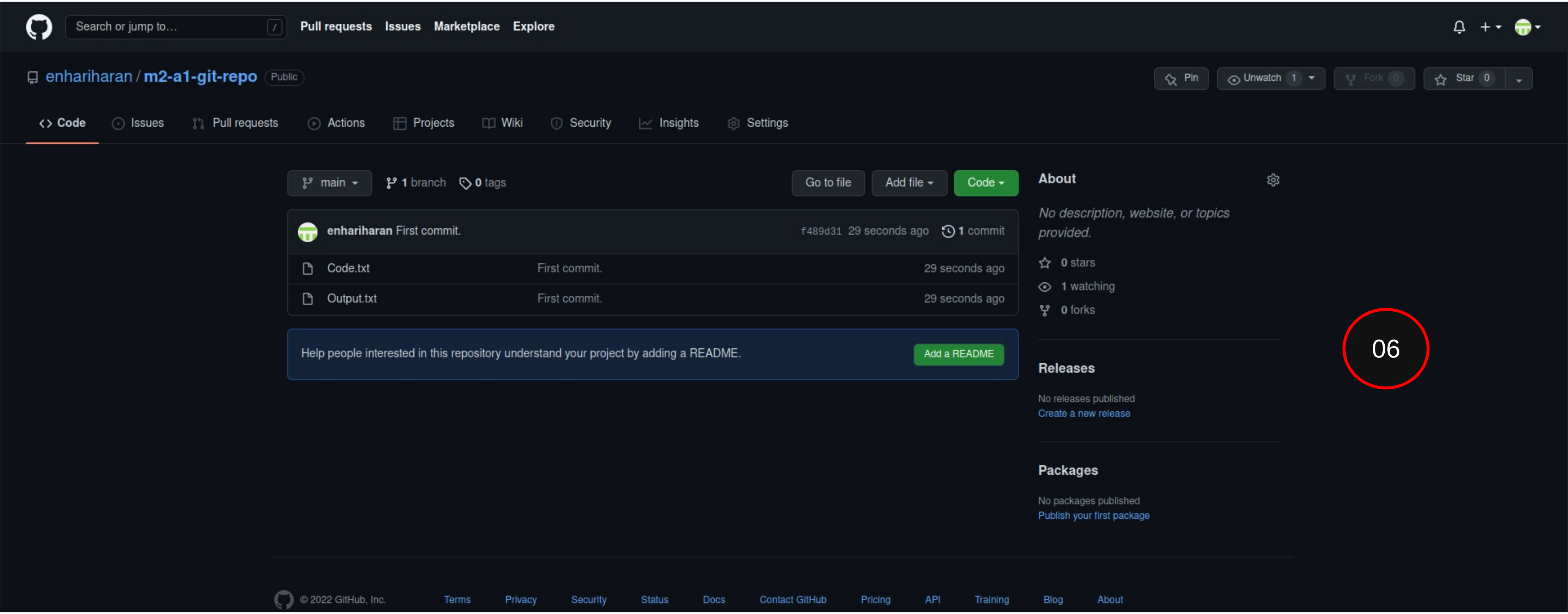


```
(base) hariharan@hariharan-home-pc:~$ mkdir m2-a1-git-repo/
(base) hariharan@hariharan-home-pc:~$ cd m2-a1-git-repo/
(base) hariharan@hariharan-home-pc:~/m2-a1-git-repo$ touch Code.txt Log.txt Output.txt
(base) hariharan@hariharan-home-pc:~/m2-a1-git-repo$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/hariharan/m2-a1-git-repo/.git/
(base) hariharan@hariharan-home-pc:~/m2-a1-git-repo$ git branch -m main
(base) hariharan@hariharan-home-pc:~/m2-a1-git-repo$ git stage Code.txt Output.txt
(base) hariharan@hariharan-home-pc:~/m2-a1-git-repo$ git commit -m "First commit."
[main (root-commit) f489d31] First commit.
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Code.txt
 create mode 100644 Output.txt
(base) hariharan@hariharan-home-pc:~/m2-a1-git-repo$ git remote add origin git@github.com:enhariharan/m2-a1-git-repo.git
(base) hariharan@hariharan-home-pc:~/m2-a1-git-repo$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 239 bytes | 239.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:enhariharan/m2-a1-git-repo.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
(base) hariharan@hariharan-home-pc:~/m2-a1-git-repo$ 
```

05

5. Execute the commands in the order shown to:

a. Create a new directory “m2-a1-git-repo” and go into it
b. Create empty files “Code.txt”, “Log.txt”, and “Output.txt”
c. Initialize a Git repository in this directory
d. Rename the default “master” branch created to “main”
e. Stage the files “Code.txt” and “Output.txt”
f. Commit them
g. Create a remote “origin” that points to the GitHub repository created in (3)
h. And finally push the local repo to github



enhariharan / m2-a1-git-repo Public

enhariharan First commit. f489d31 29 seconds ago 1 commit

Code.txt	First commit.	29 seconds ago
Output.txt	First commit.	29 seconds ago

Help people interested in this repository understand your project by adding a README.

About

No description, website, or topics provided.

0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

© 2022 GitHub, Inc.

06

6. Refresh the repo in Github to verify that the files were pushed correctly.