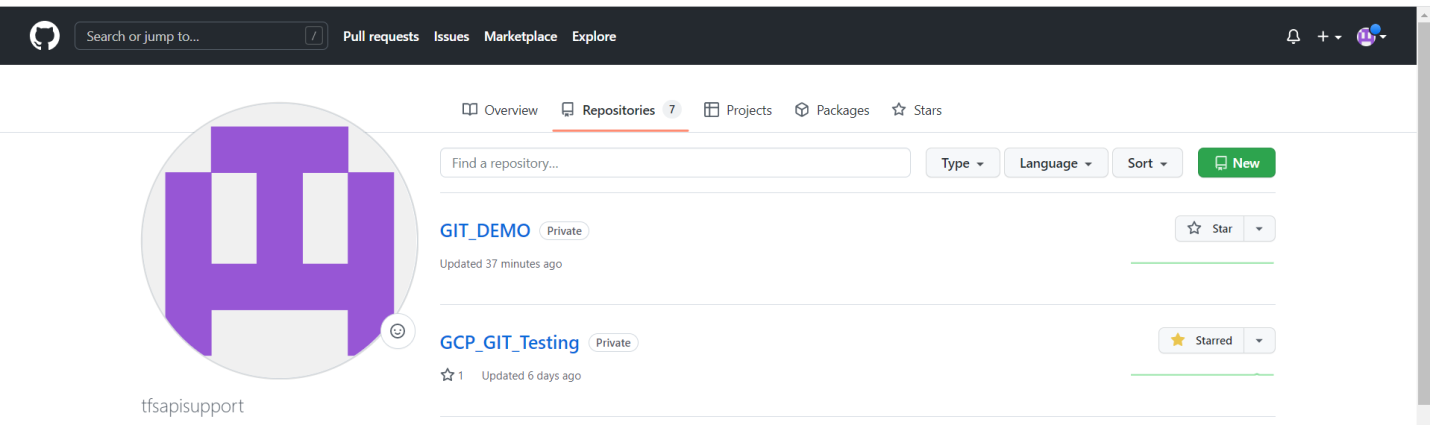
**Git Branch Creation**

* Log into the Git Hub using login url and select the Repository

<https://github.com/login>

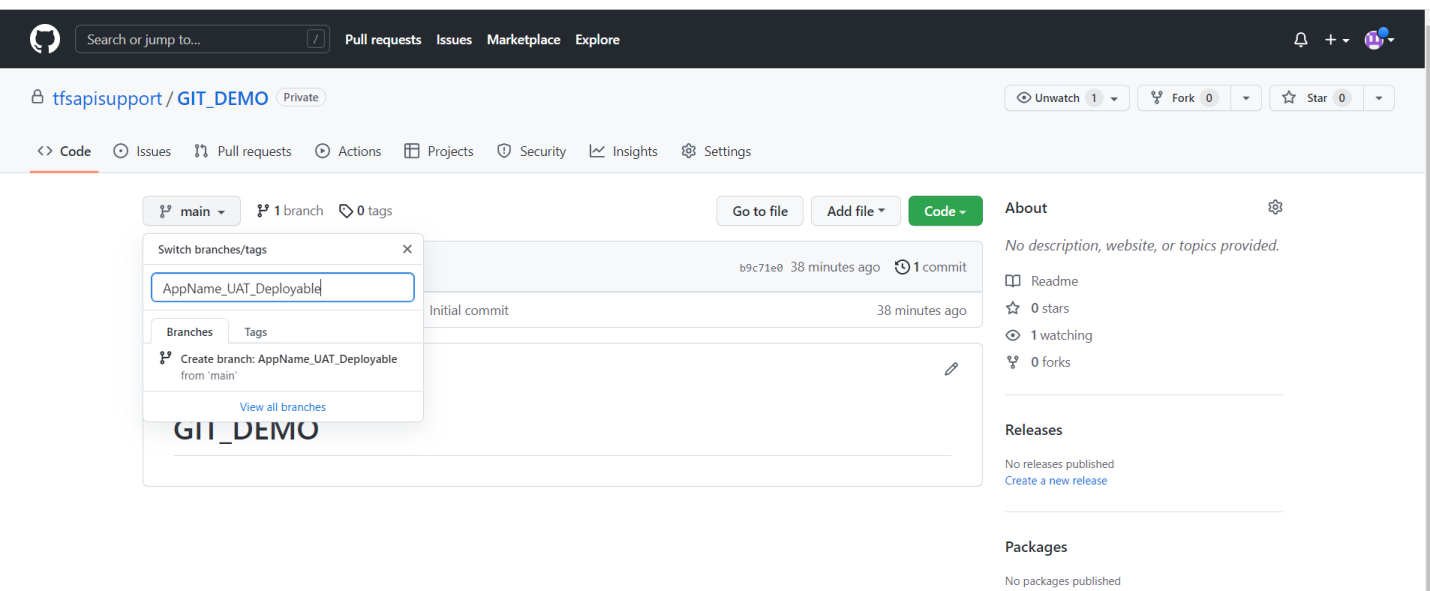


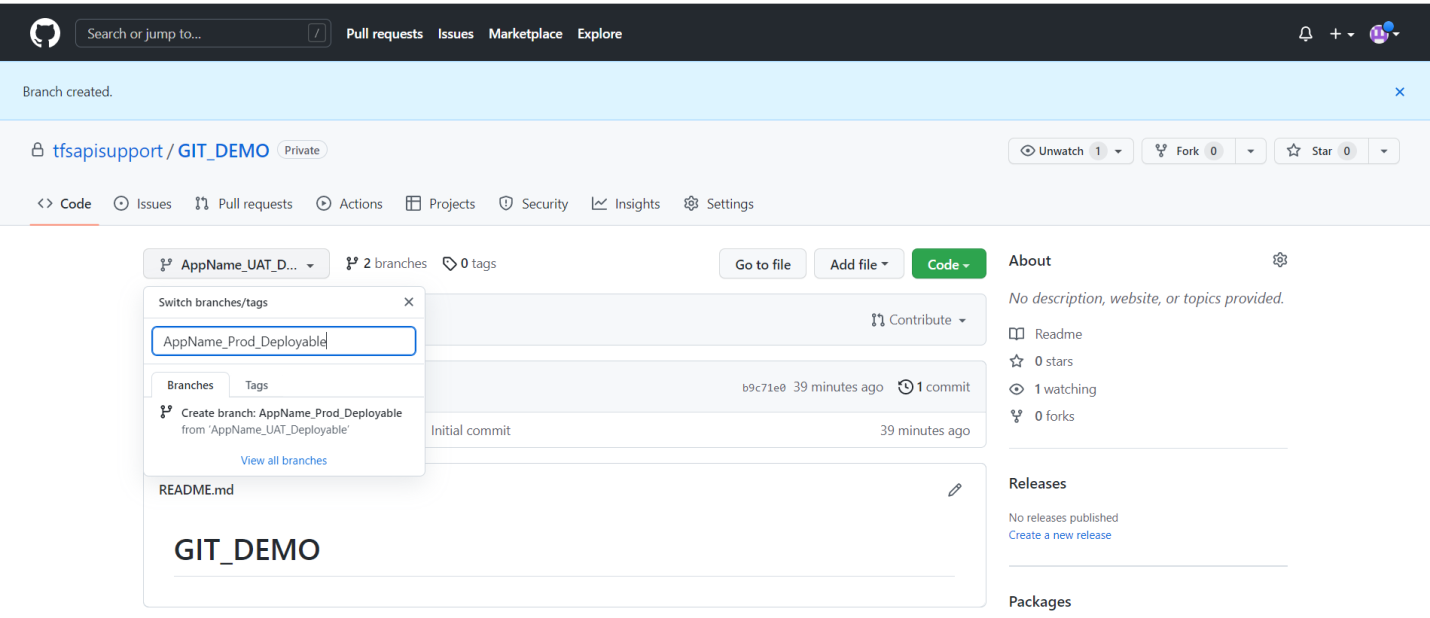
* Create a new branch as shown which fulfills the below requirement click on create

**ApplicationName\_Environment\_Deployable**

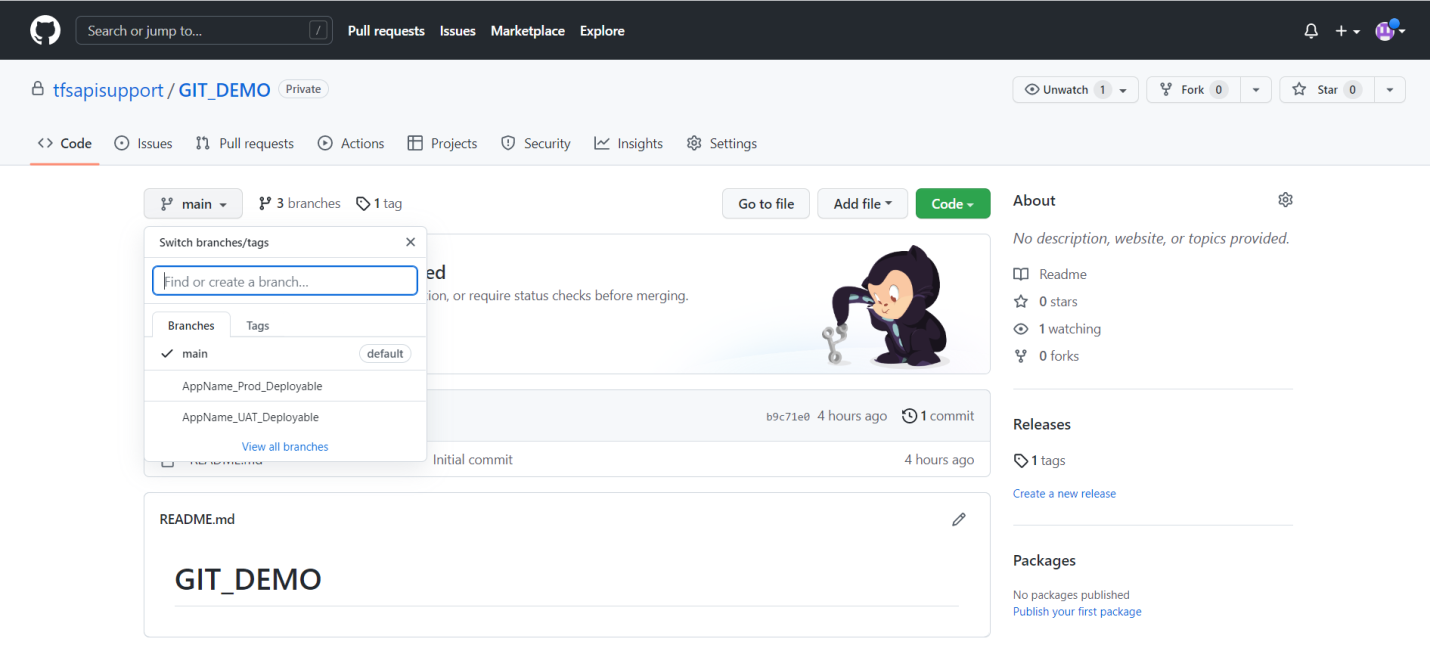
Eg: AppName\_UAT\_Deployable

Eg: AppName\_Prod\_Deployable



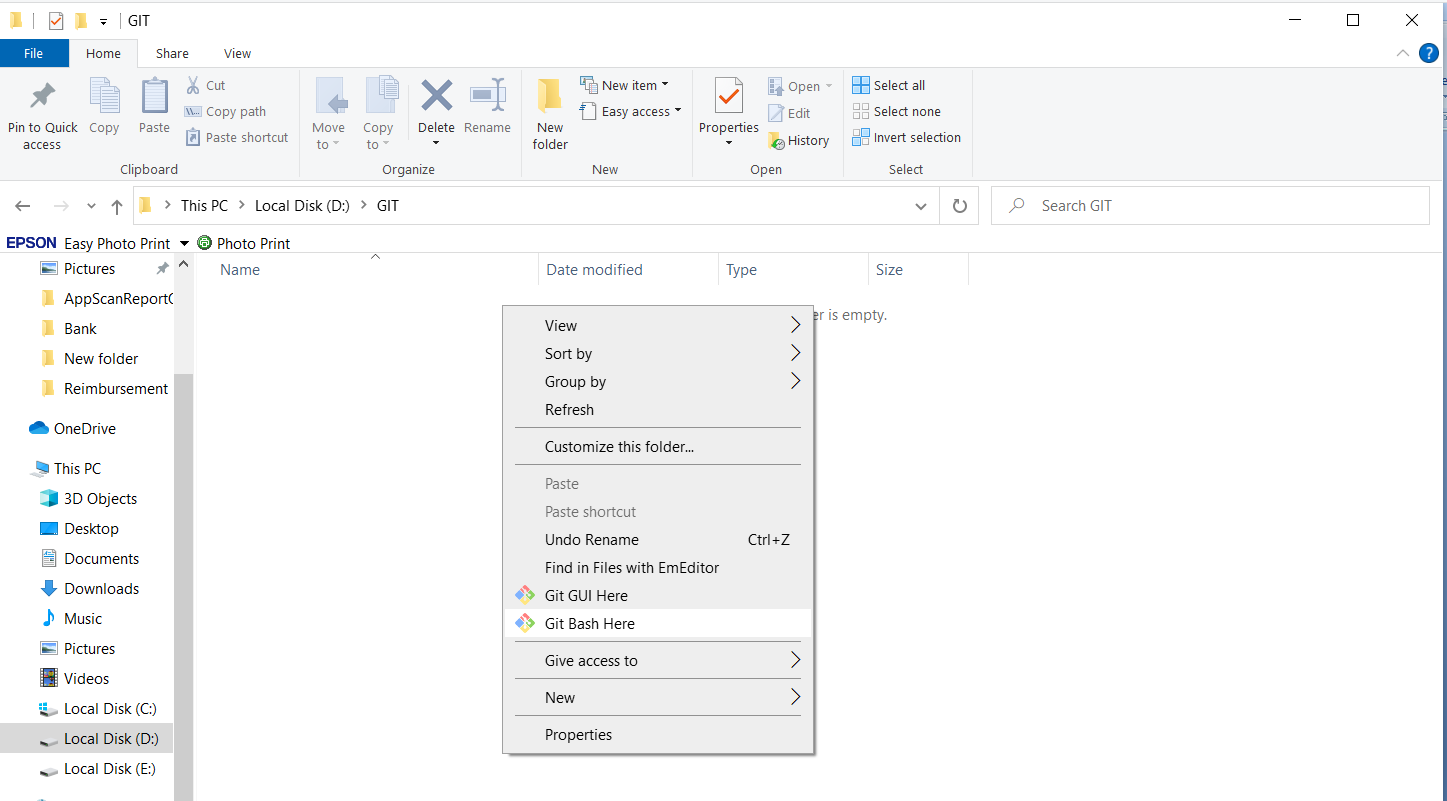


* Check the created branches by clicking the dropdown button on main branch & can switch to the created branches by clicking from the list below.

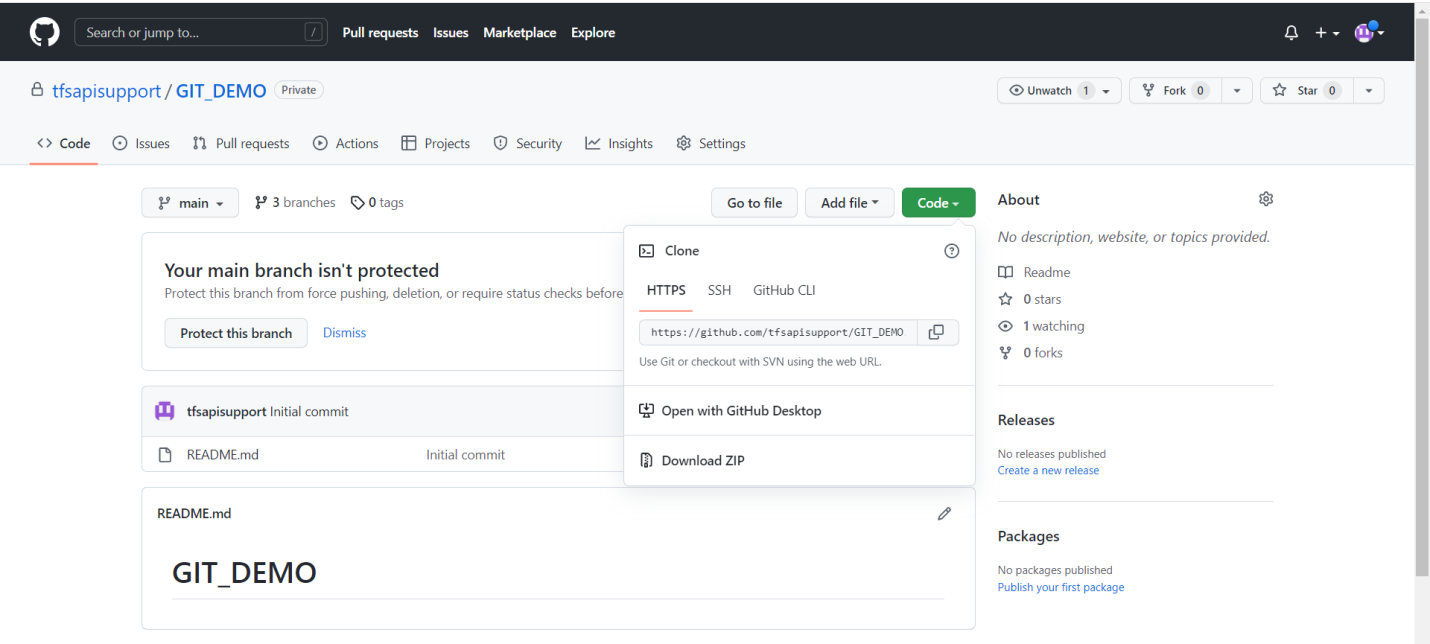


**Git Cloning Local Machine**

* Download and install git bash before proceeding further
* Create a new folder in local machine to clone the remote repository and open git bash from that location as shown below

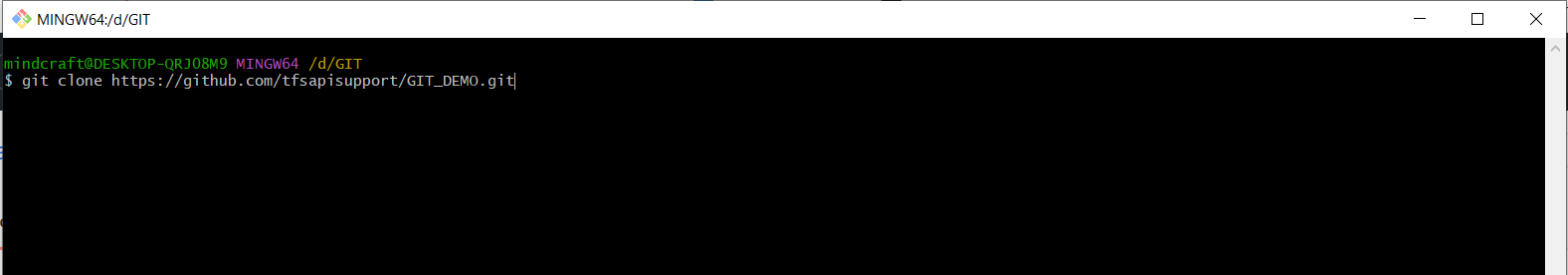


* Copy the **Git URL** from the remote repository as shown below

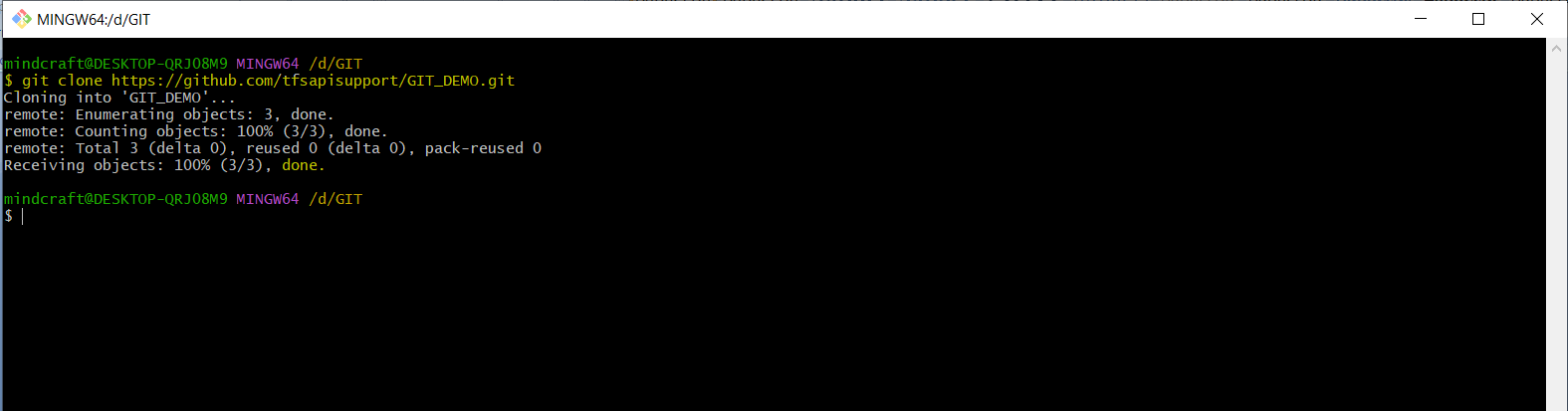


* Run the below command from the git bash to clone

git clone **copied Git URL**



Output:



* Move to the remote repository which was cloned in local machine as shown.

**cd RepoName**



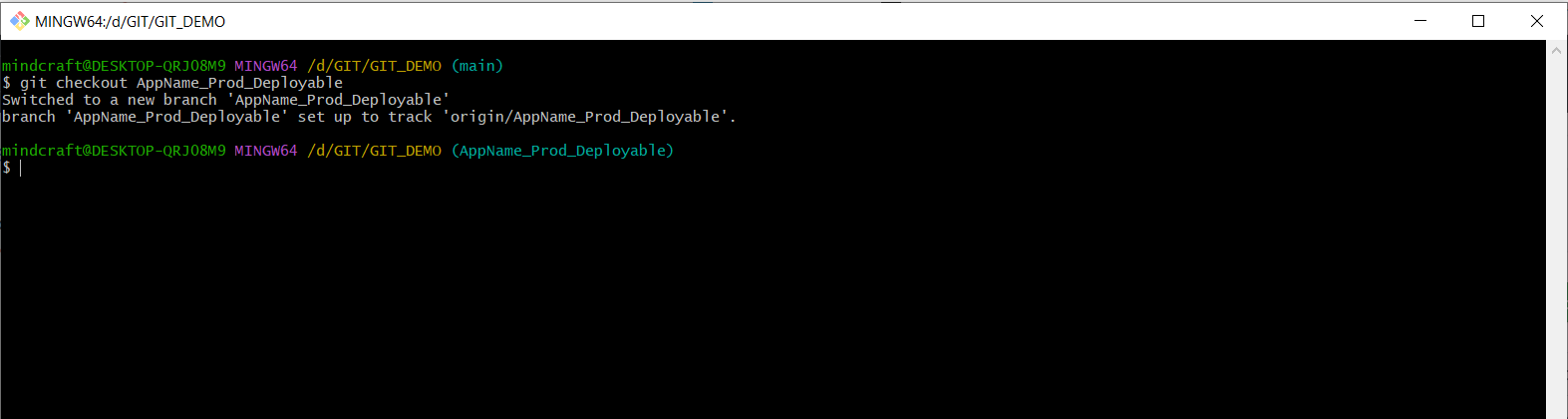
* Run the below command to check on which branch currently we are in

**git branch -a**

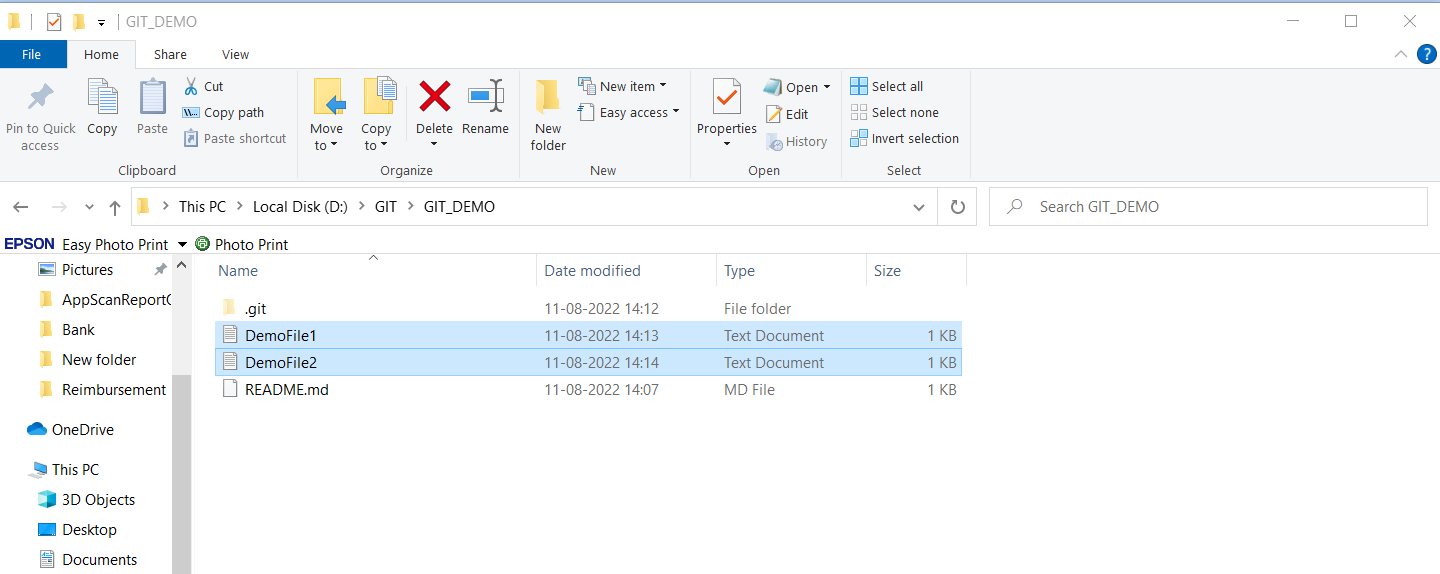


* Run below command to checkout to the branch created “AppName\_Prod\_Deployable”

**git checkout BranchName**



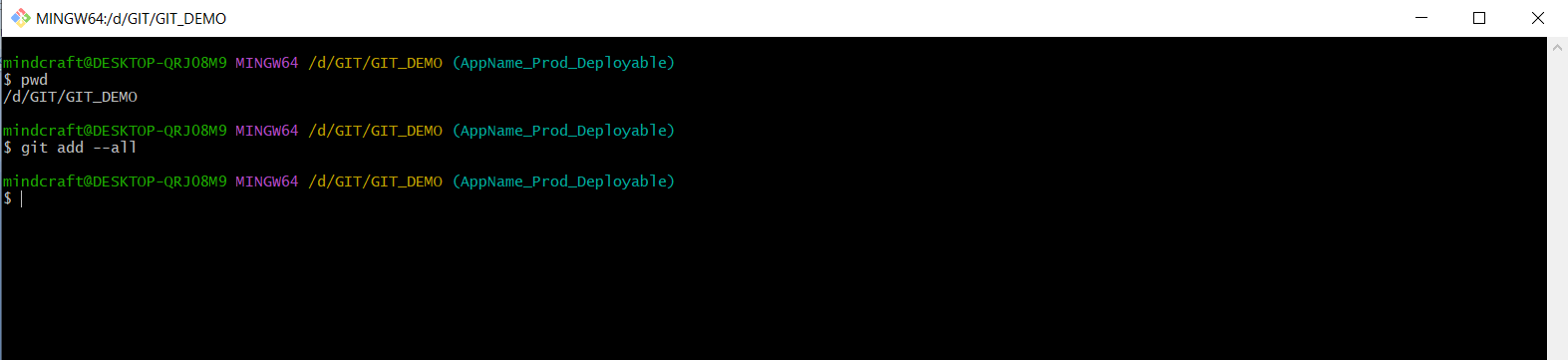
* Open the remote repository cloned in local machine & add the deployable files in local machine as shown.



* Run the below command from git bash to add files.

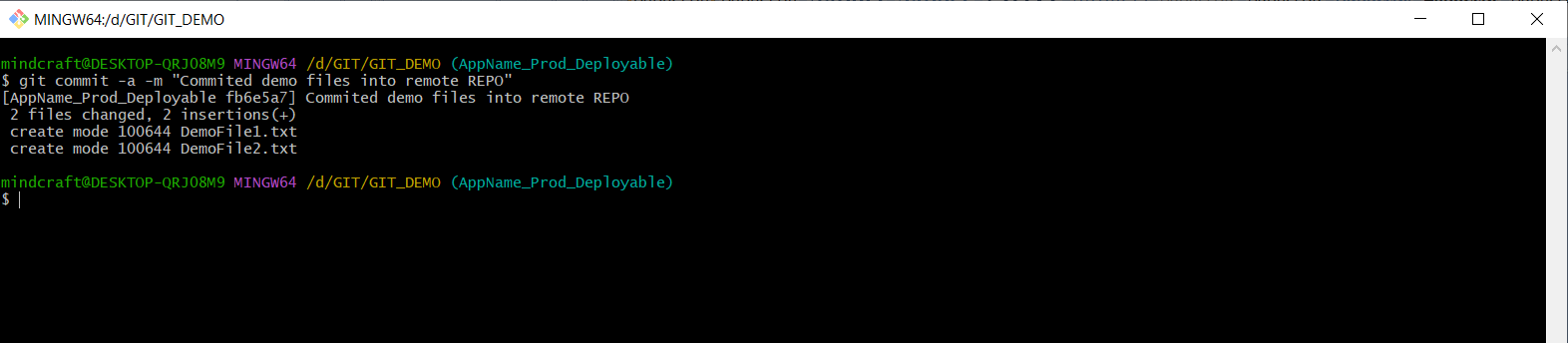
git add --all <to add all files at once>

git add <file name OR folder name>



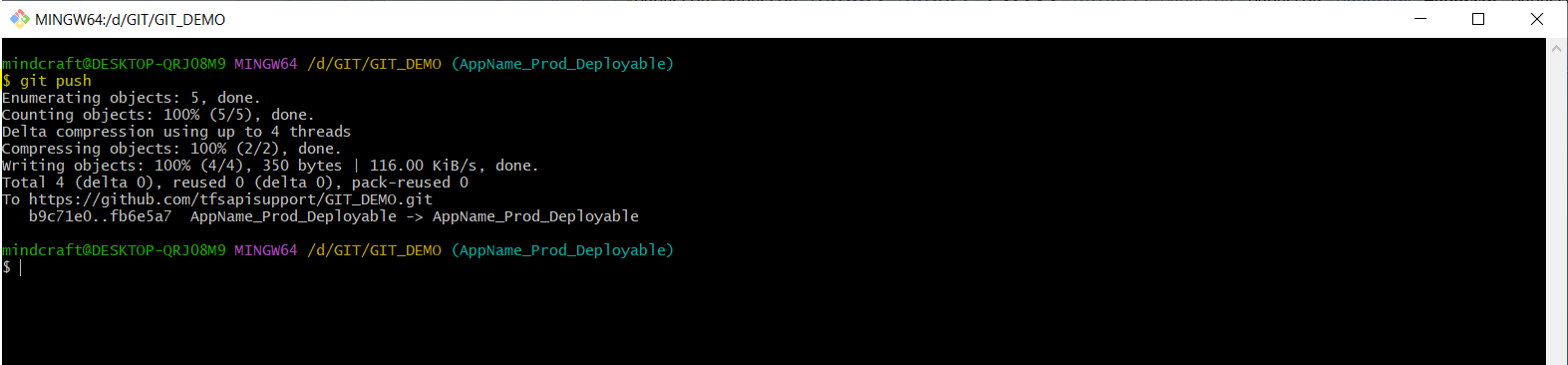
* Run the below command to add a comment to the added files.

git commit -a -m "commit message"

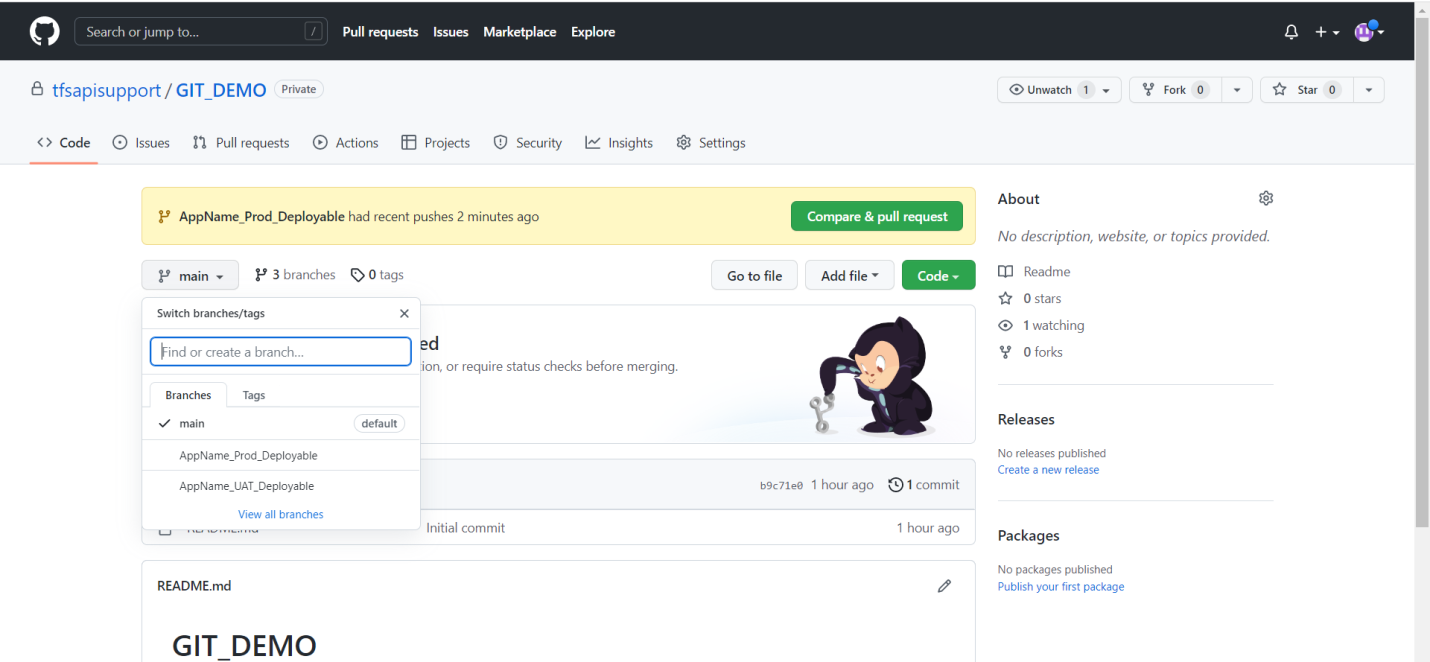


* Run the below command to push the deployable files into remote repository.

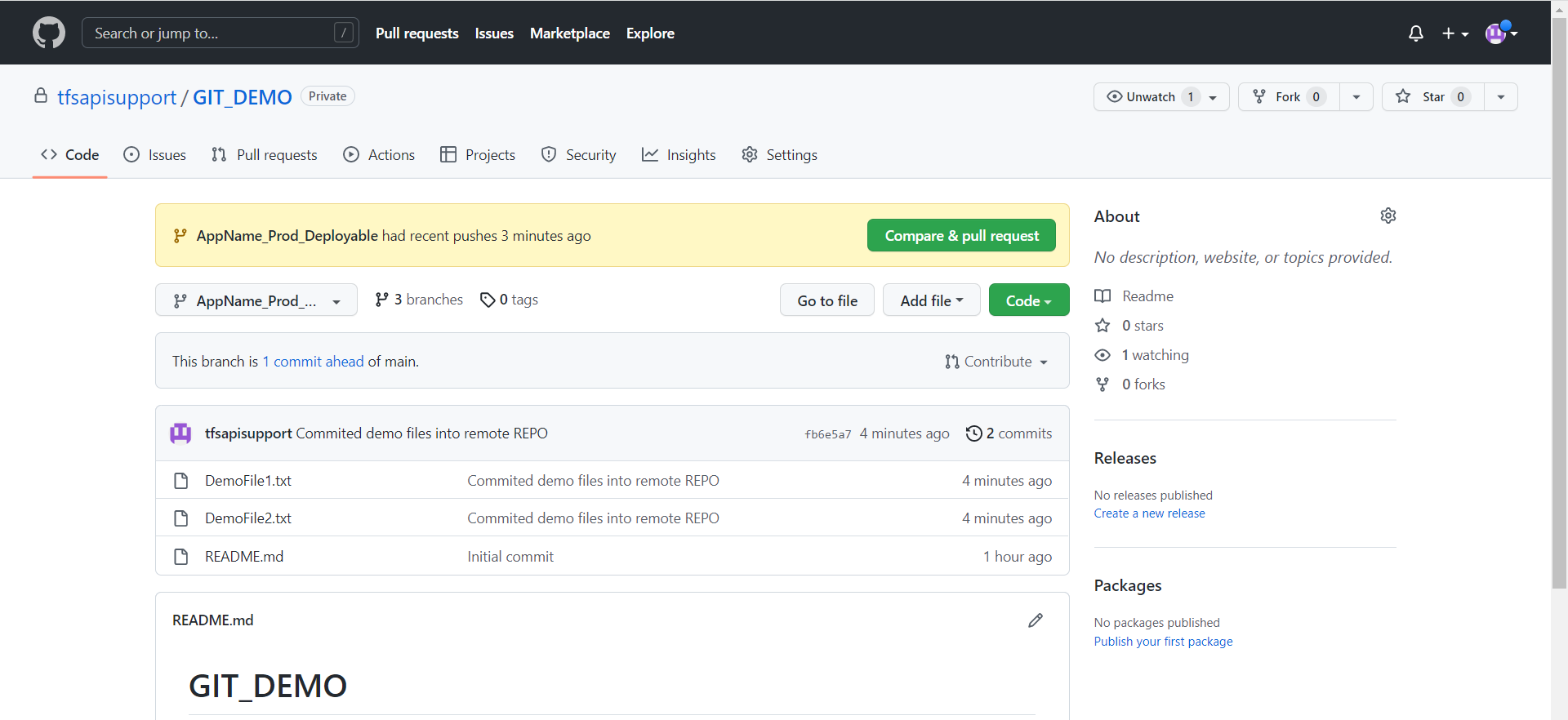
git push



* Open the remote repository in browser, refresh the page & change the branch from main to “AppName\_Prod\_Deployable”



* You can see the deployable files added in the remote repository



**TAG / Version Controlling System**

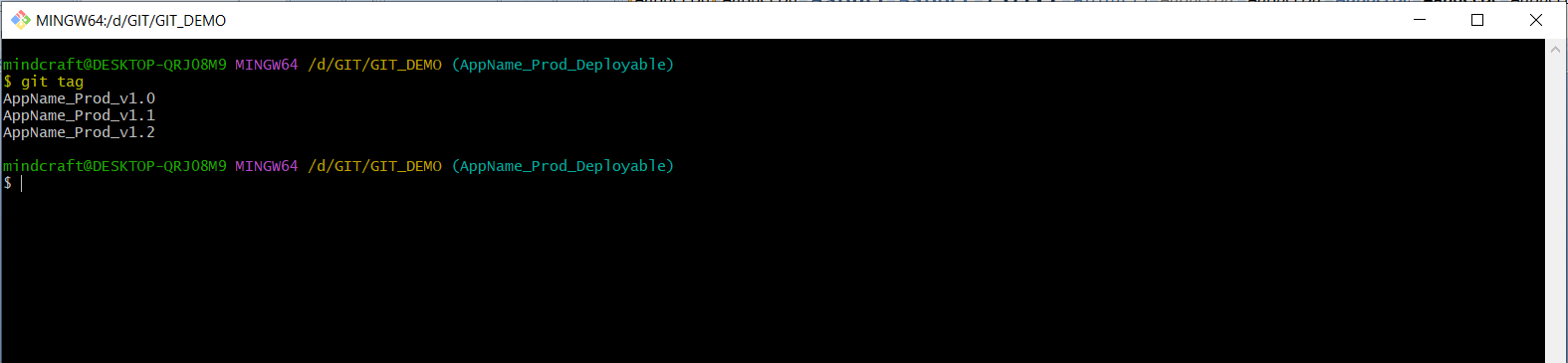
* Run the below command to add the VFS

git tag **AppName\_Environment\_versionNum**



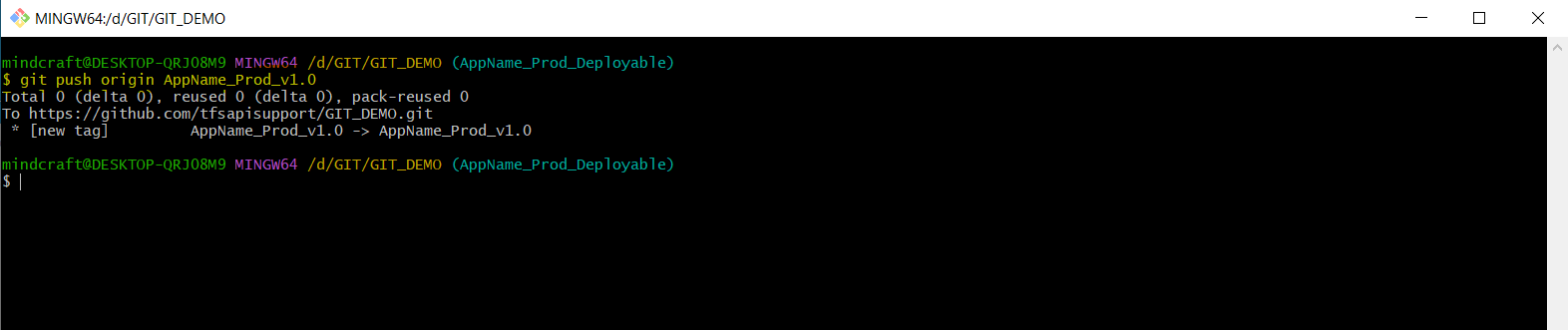
* Run the below command to check the current number of versions available.

git tag

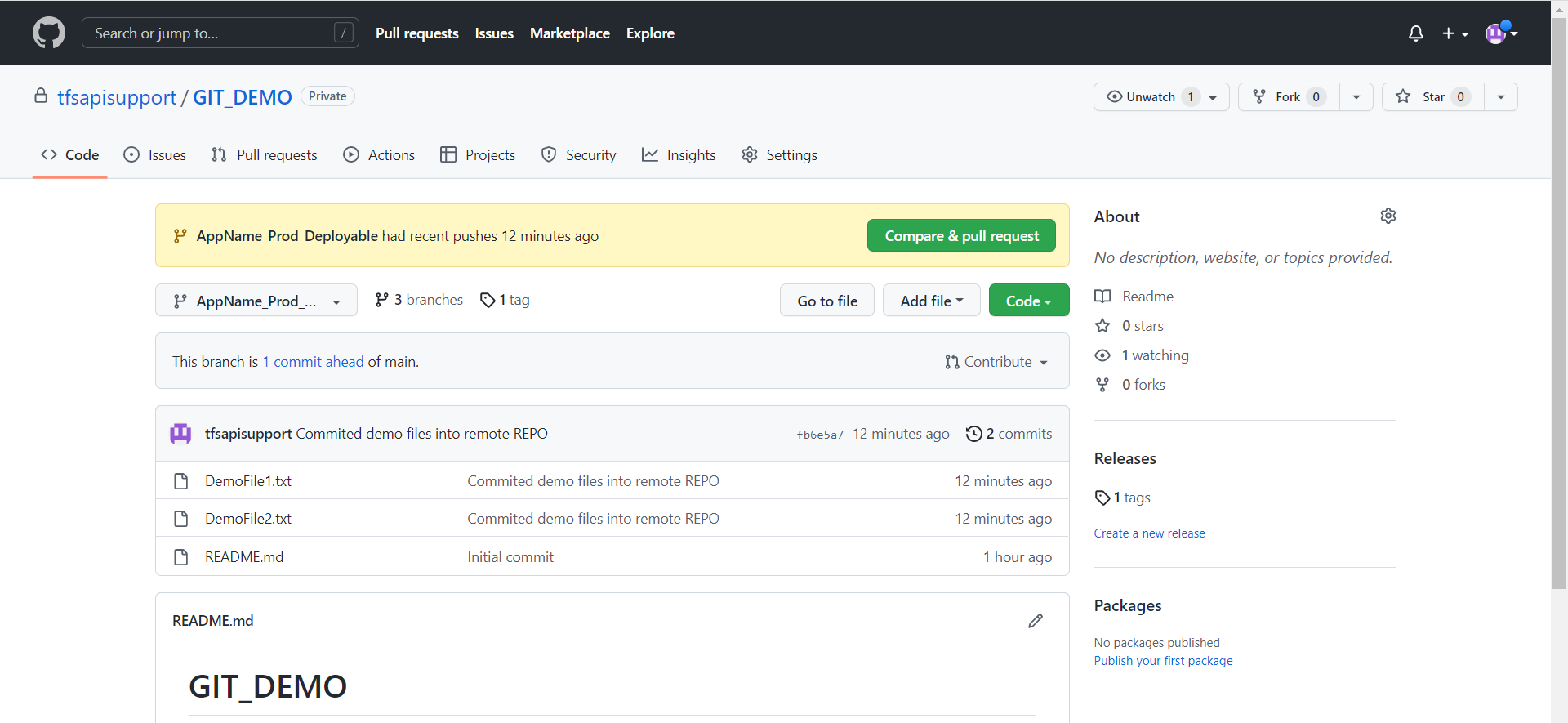


* Run the below command to push the VFS to remote repository.

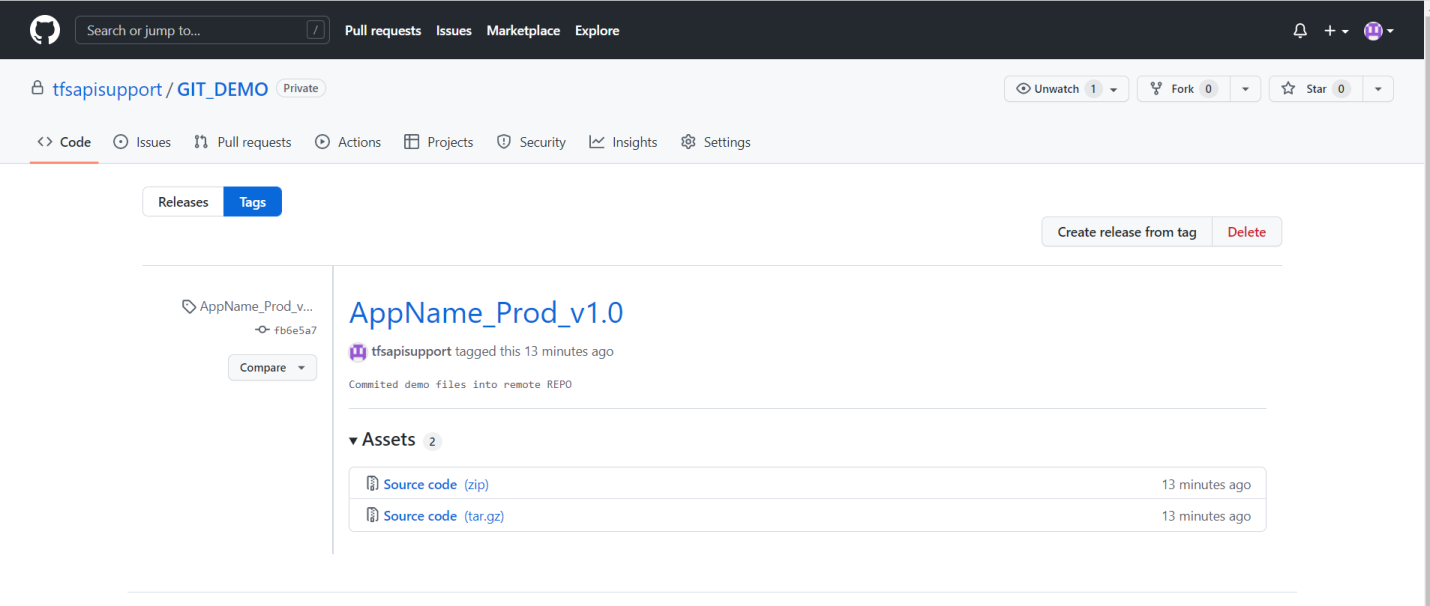
git push origin **AppName\_Environment\_versionNum**



* Open the remote repository in browser& refresh the page to see the list of versions.



* Open the tag & download the zip files to check the deployable files added under that version.

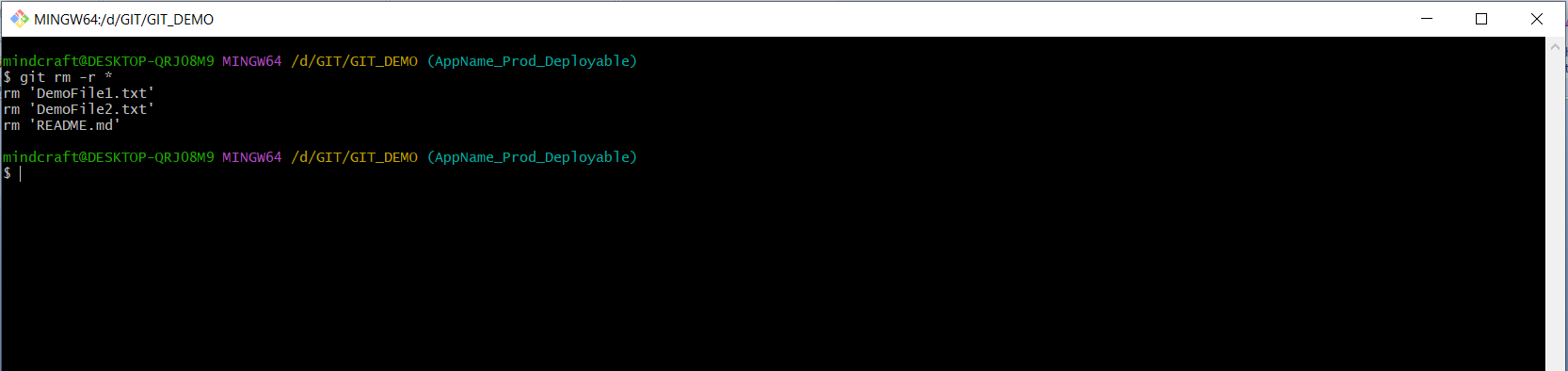


NOTE: Remove all the files from branch, once a TAG/Version is successfully created & pushed into remote repository.

**Deleting the deployable files from branches**

* Run the below command from git bash & remove the current file before adding a new files & creating a new version.

git rm -r \*

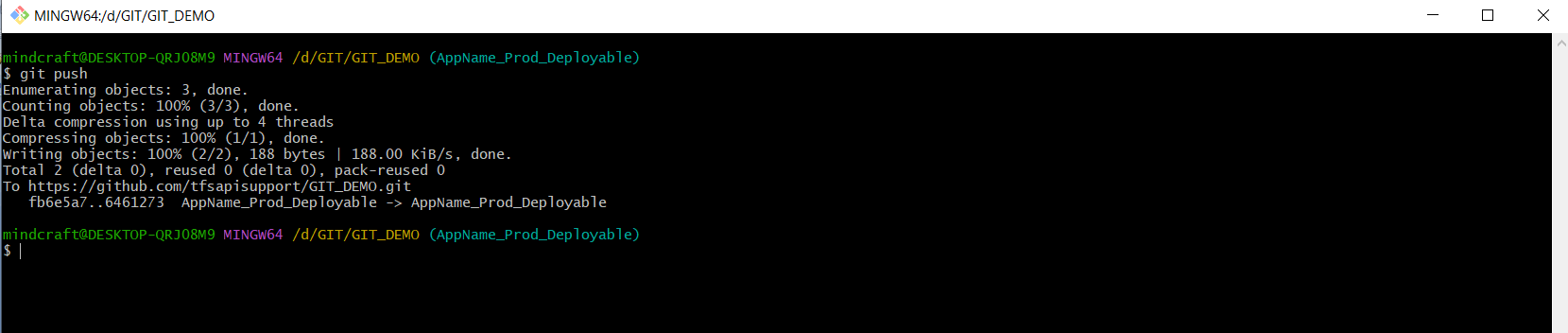


* Commit these empty changes into the remote repository by the below command

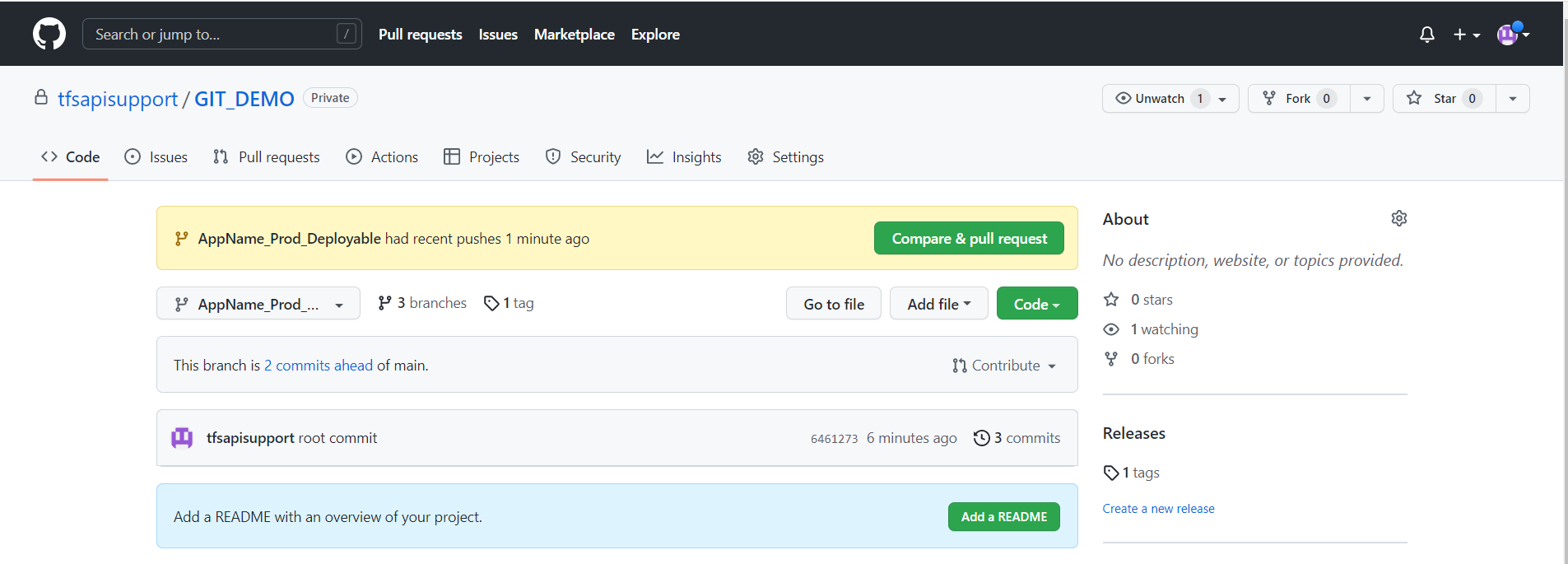
git commit --allow-empty -m "root commit"



* Push the empty changes to the remote repository with the below command.



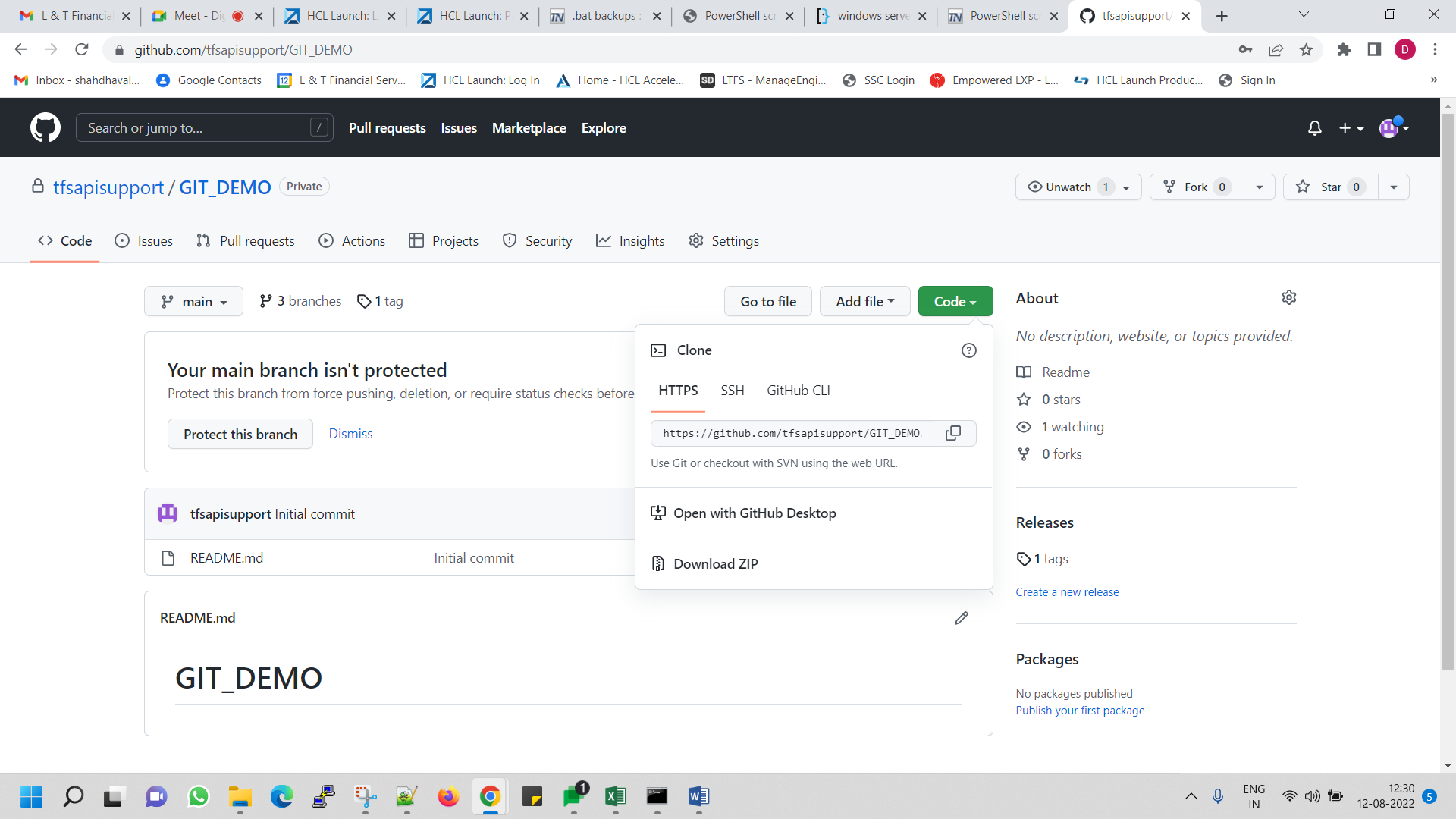
* Check the branch will be empty from browser



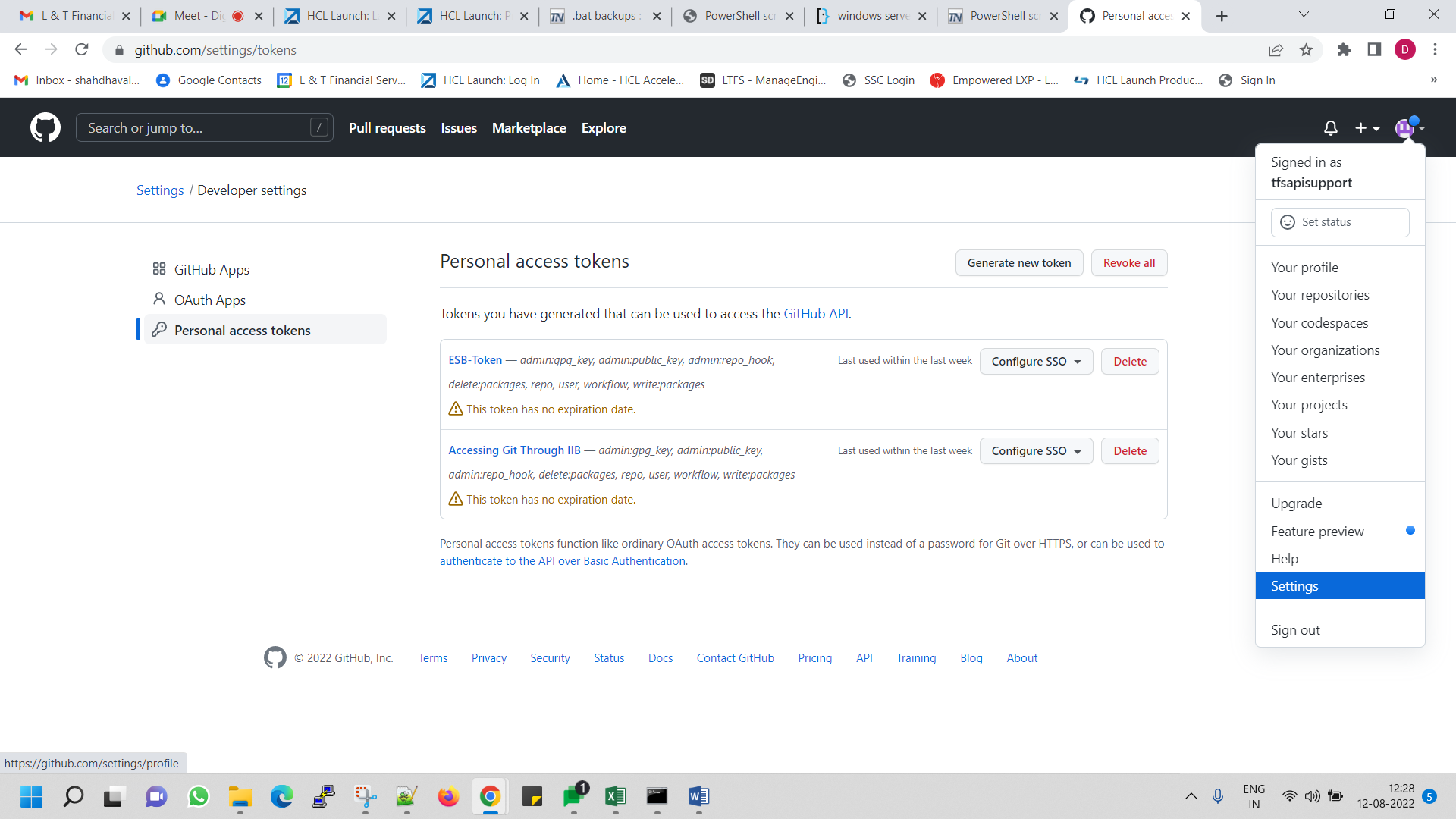
Generate Access token with Full permission and share to Devops Team with Below details

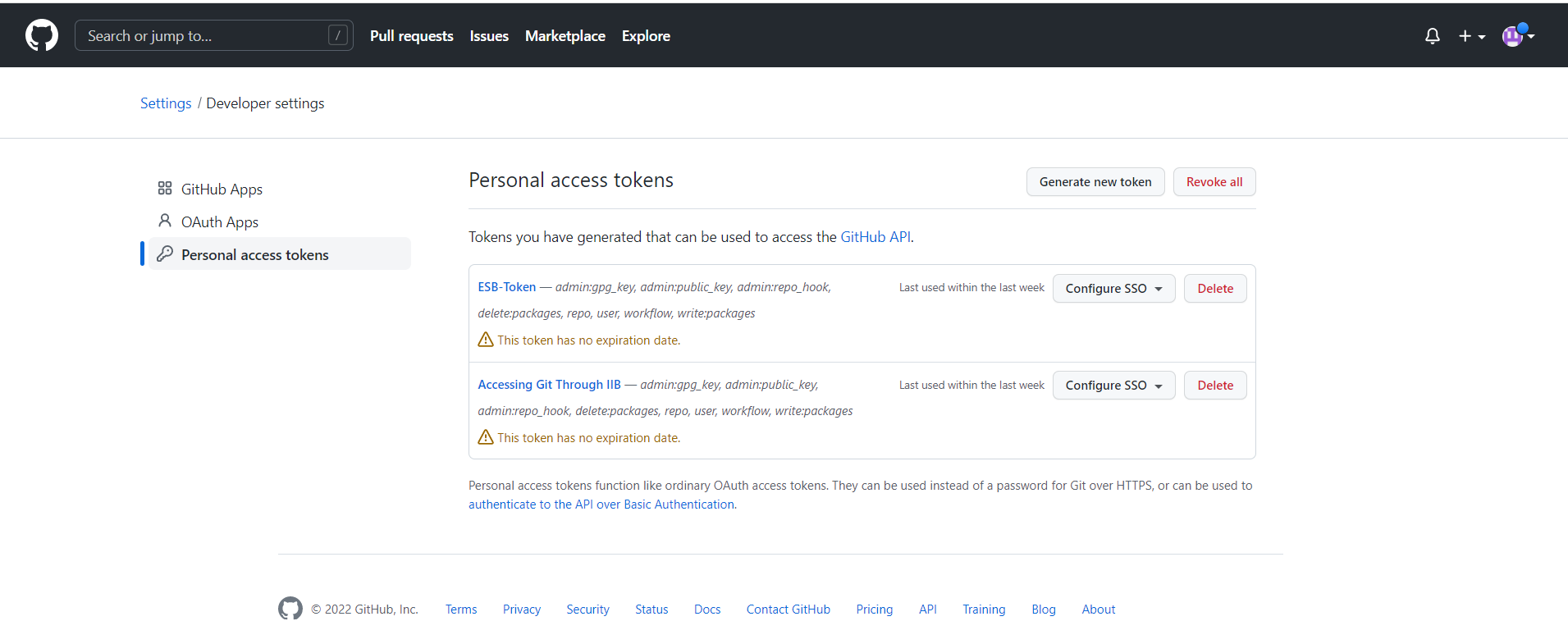
1. Username
2. Access token
3. GIT CODE URL

GIT CODE URL:



Access token: -





Select all Check box with (read/write) access while creating Generate New token also select “No expiration” in expiration drop down.

