TECHLAKE
CLOUD IS THE FUTURE.

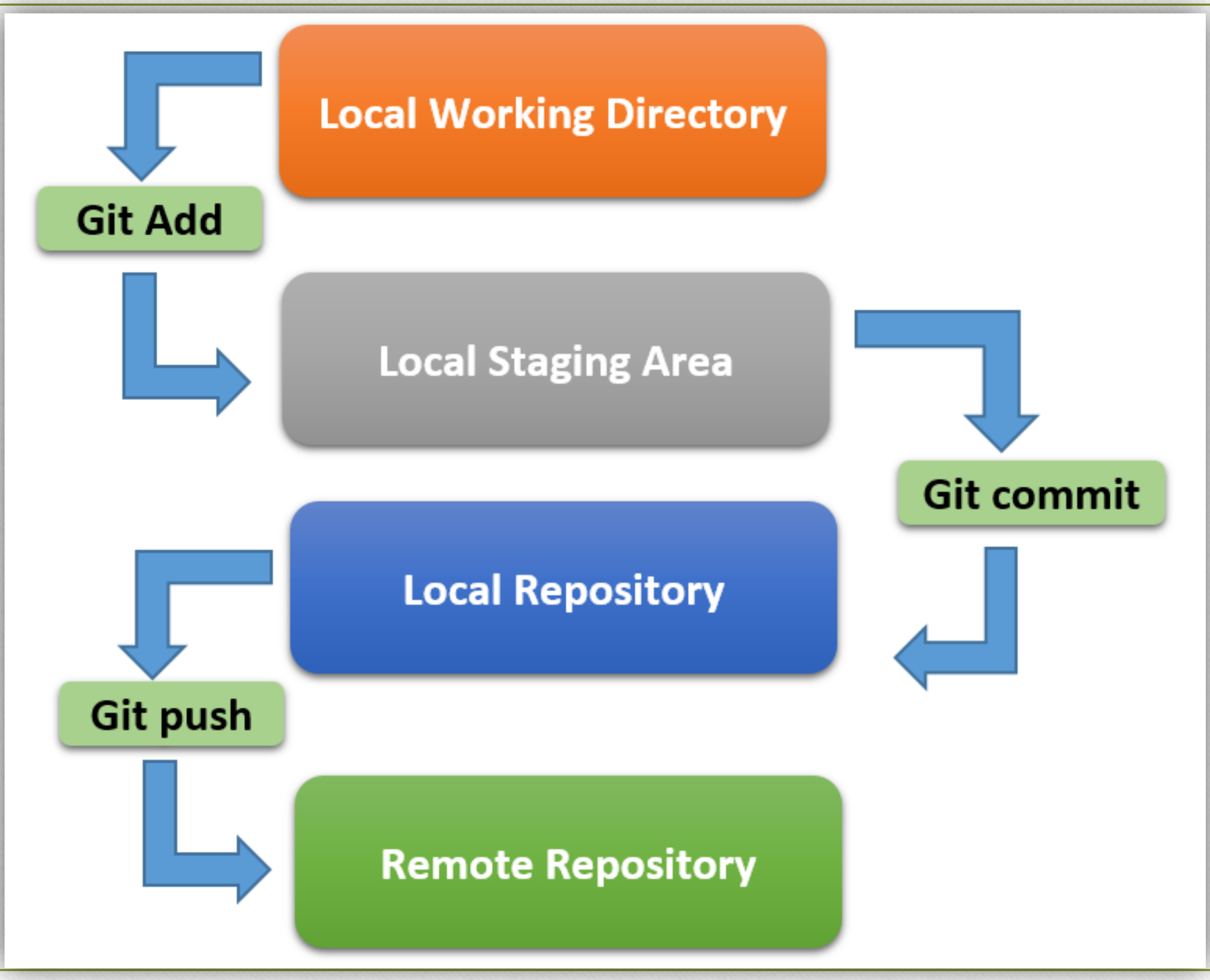
@trraveendra

TR Raveendra

git

Git Basic Tutorial

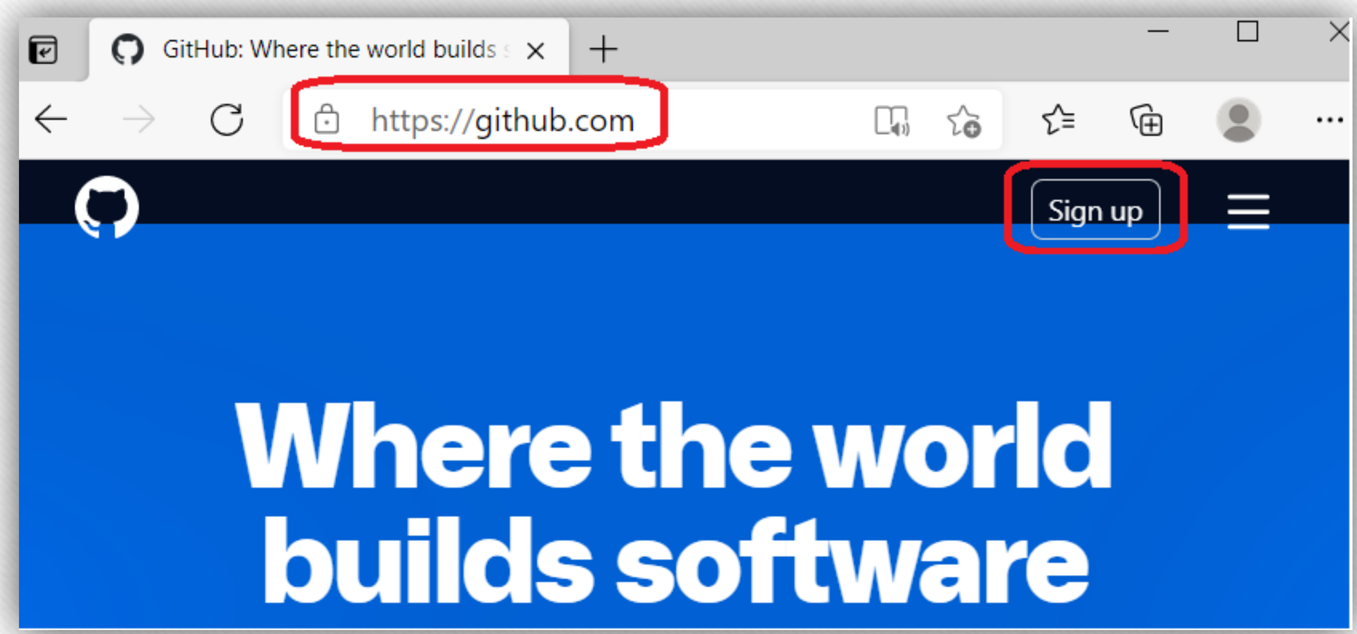
Source: github.com and git-scm.com


Local Working Directory
Git Add
Local Staging Area
Git commit
Local Repository
Git push
Remote Repository

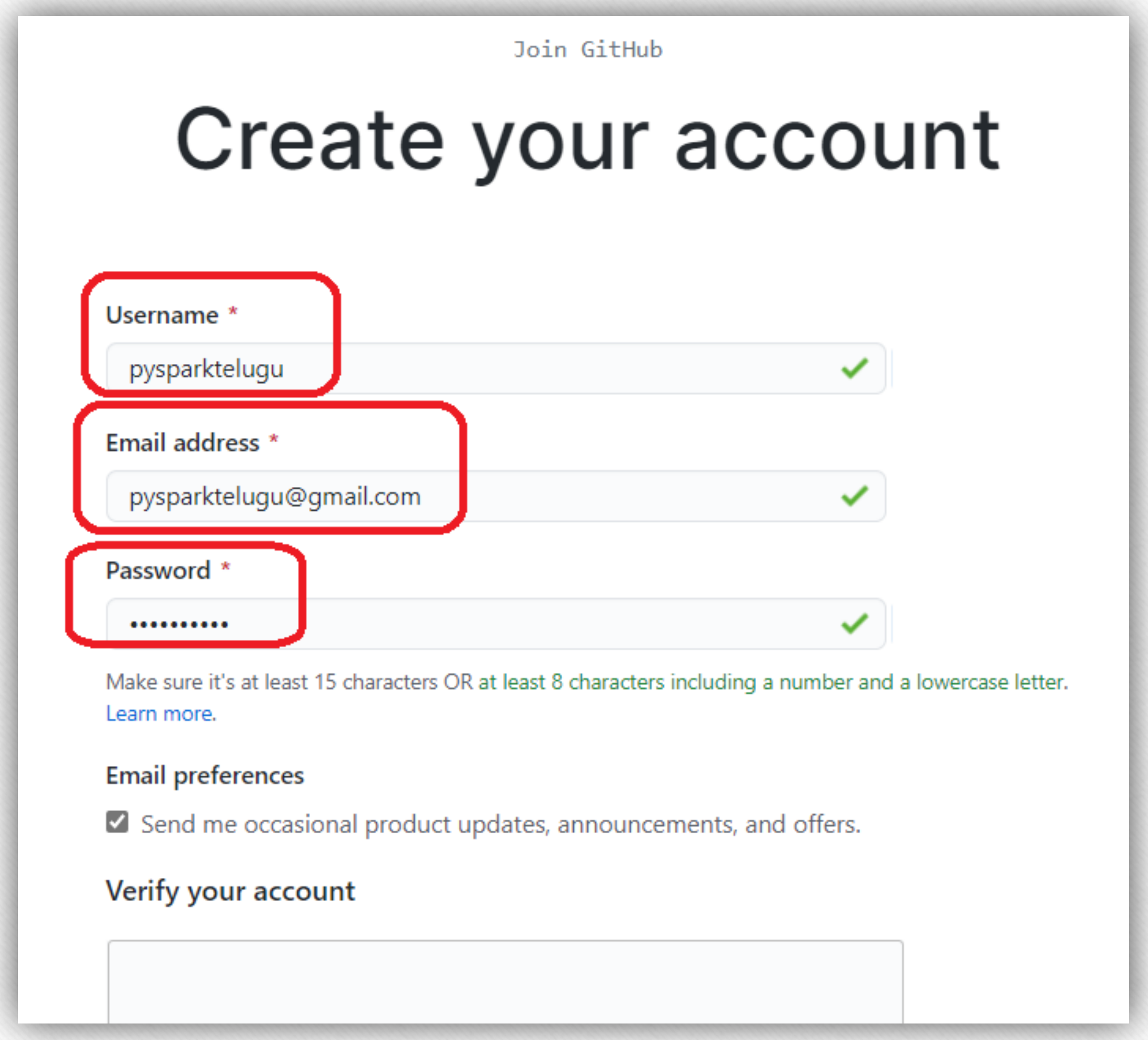
git help: possibly the most useful Git command, this command allows you to search the most common Git commands in the command/terminal shell. If you follow this command with another command or concept (i.e.. git help push) Git generates a html page detailing the command or concept as well as possible options for its use.
git init: initializes a git repository by creating the initial .git directory in a new or in an existing project.
git clone: copies an existing GitHub repo to local machine.
git status: checks the working directory to see if up-to-date with the remote repo.
git add: a.k.a. staging changes, adds changes to staging area of the working directory. This command is the first step in committing changes to your local version of the repo before pushing them to the remote GitHub version of the repo
git commit: tells Git to record the changes made to your version of the repo. Every commit needs to have a message that explains what files have been edited/added. After the command add -m and then the commit message in quotes (git commit -m "This is where your message goes").
git pull: this command is made up of two other commands (**git fetch** and **git merge**) and is used to fetch the data from a remote repository and merge it into your local computer's version of the repository. If working in a fork it is important to remember that changes are pulled from the original remote repo the fork is made from: git pull upstream master

git push: updates remote repos to match commits made on local machine. If working in specific branches you can designate the branch name (that you wish to push) after the command, or use git push -all to add commits from all local branches.
git stash: used when you want to record the current changes to the working directory, but want to go back to a clean working directory without forever losing those changes and without adding them to the staging area. This command saves your local modifications temporarily outside of your working directory and reverts the working directory to match the last commit (typically used so you can pull in remote changes to a clean directory while avoiding possible merge conflicts and keeping your changes available).
git remote: used to retrieve a list of remote repositories associated with local repository. To create a new remote association use git remote add followed by a name for the remote branch (i.e. upstream) and then the URL of the remote repo on GitHub. git remote -v displays a verbose list of the associated remote repos with the remote URL after the repo name.
git branch: used to work inside of branches.
git checkout: followed by a branch name allows you to switch into the working directory of a specific branch.
git diff: followed by the names of two branches allows you to compare the differences between the two branches. For example: git diff master development will show the differences between the master branch and the development branch

Creating Github account using gmail.
Goto www.github.com and click on Sign Up


GitHub: Where the world builds
https://github.com
Sign up
Where the world builds software

Enter User Name
Email Address
Password



Join GitHub

Create your account

Username *
pysparktelugu ✓

Email address *
pysparktelugu@gmail.com ✓

Password *
•••••••••• ✓

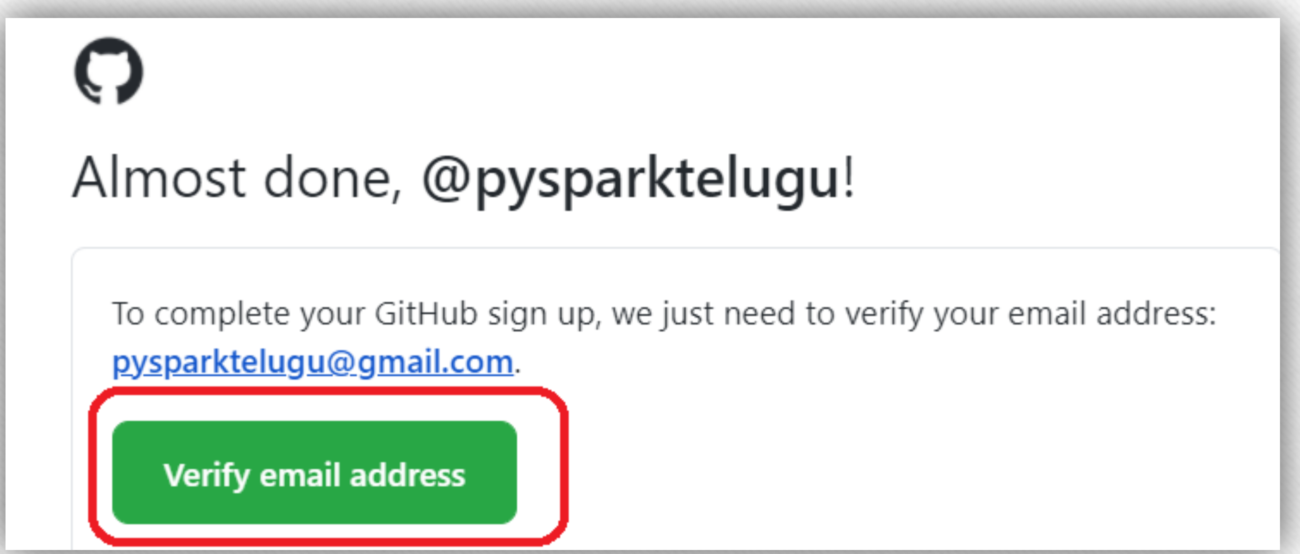
Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. Learn more.

Email preferences

☑ Send me occasional product updates, announcements, and offers.

Verify your account

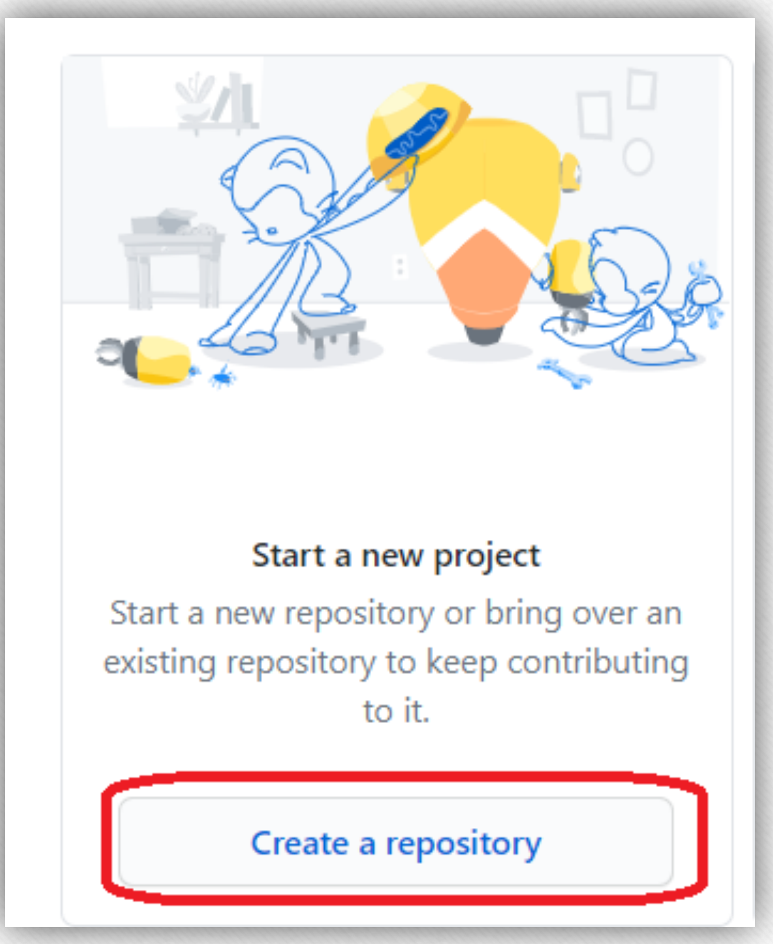
Verify Email Address. After registering you will get Email verification link to Your Gmail. Click On That Link And Verify Email ID.



Almost done, @pysparktelugu!

To complete your GitHub sign up, we just need to verify your email address: pysparktelugu@gmail.com.

Create New Repository using below option.


Start a new project

Start a new repository or bring over an existing repository to keep contributing to it.

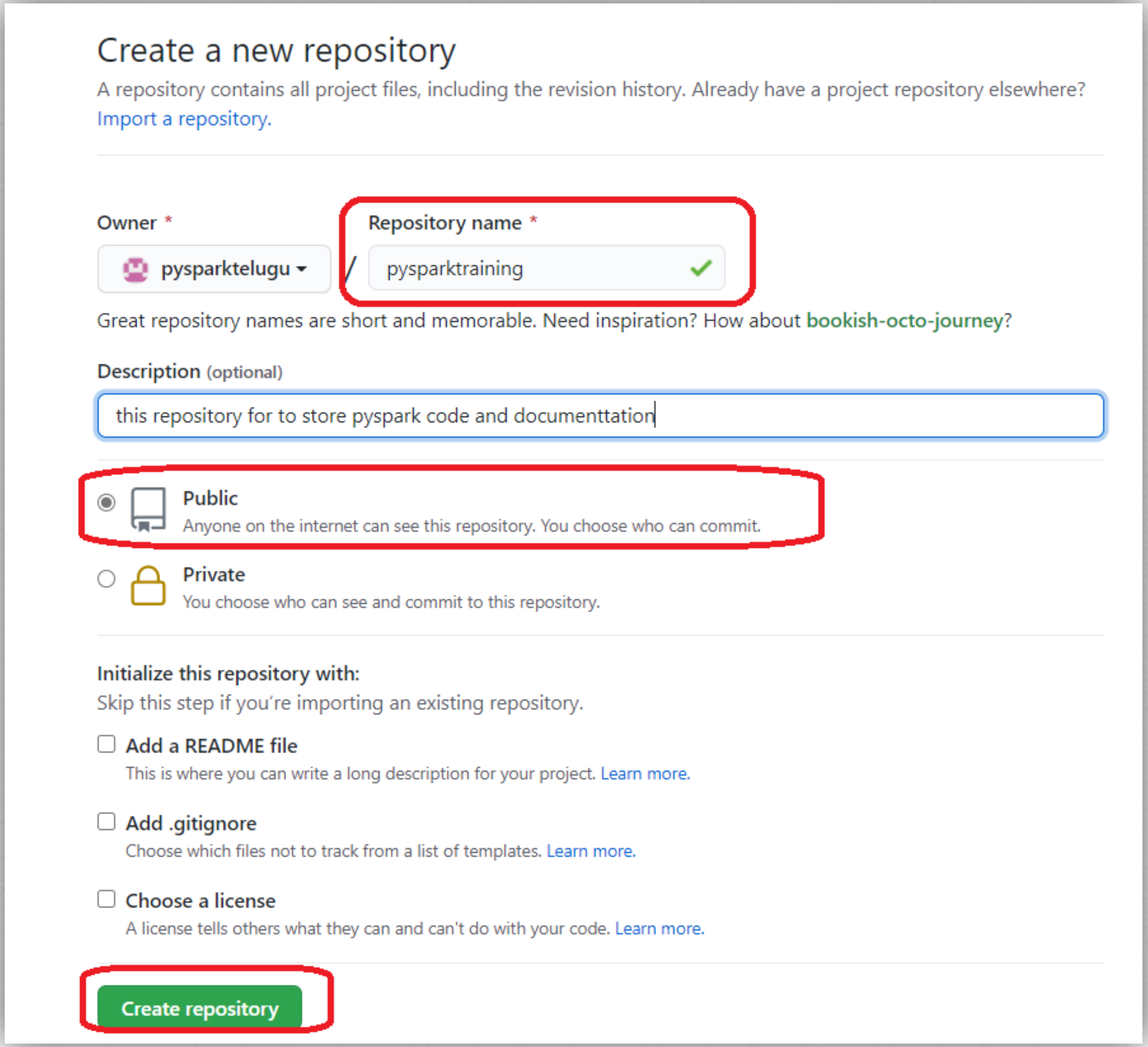
TECHLAKE
CLOUD IS THE FUTURE
@trraveendra

Enter Repository Name

Select Repository Type

Public (you can decide)

Private (Personal information & Corporate Project information)



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * pysparktelugu / Repository name * pysparktraining ✓

Great repository names are short and memorable. Need inspiration? How about bookish-octo-journey?

Description (optional)

this repository for to store pyspark code and documenttation

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

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

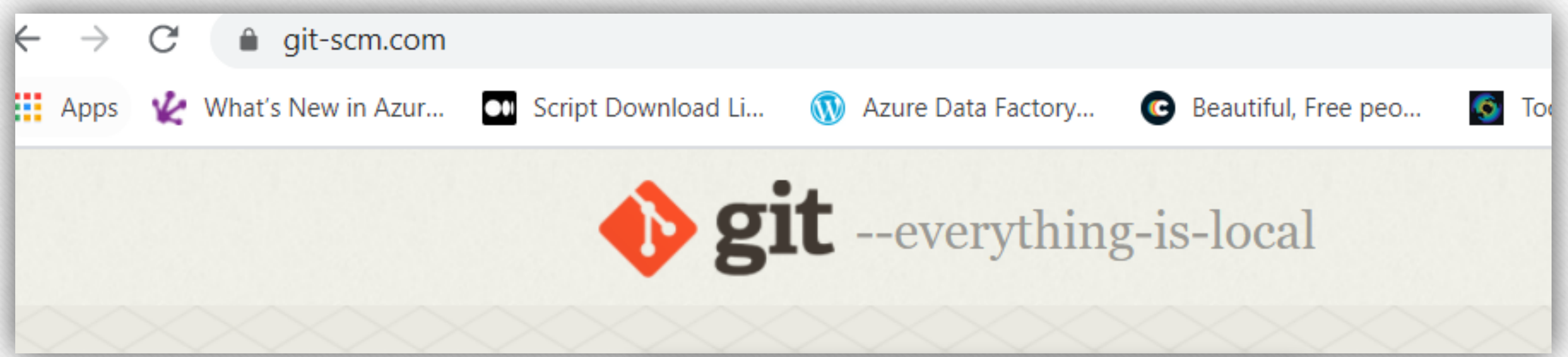
Initialize this repository with:
Skip this step if you're importing an existing repository.

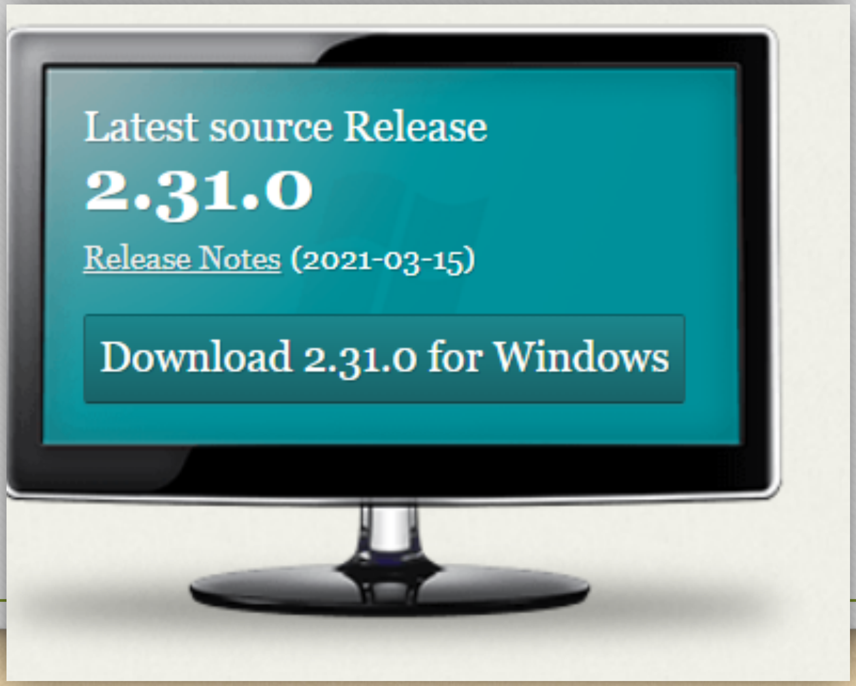
☐ Add a README file
This is where you can write a long description for your project. Learn more.

☐ 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.

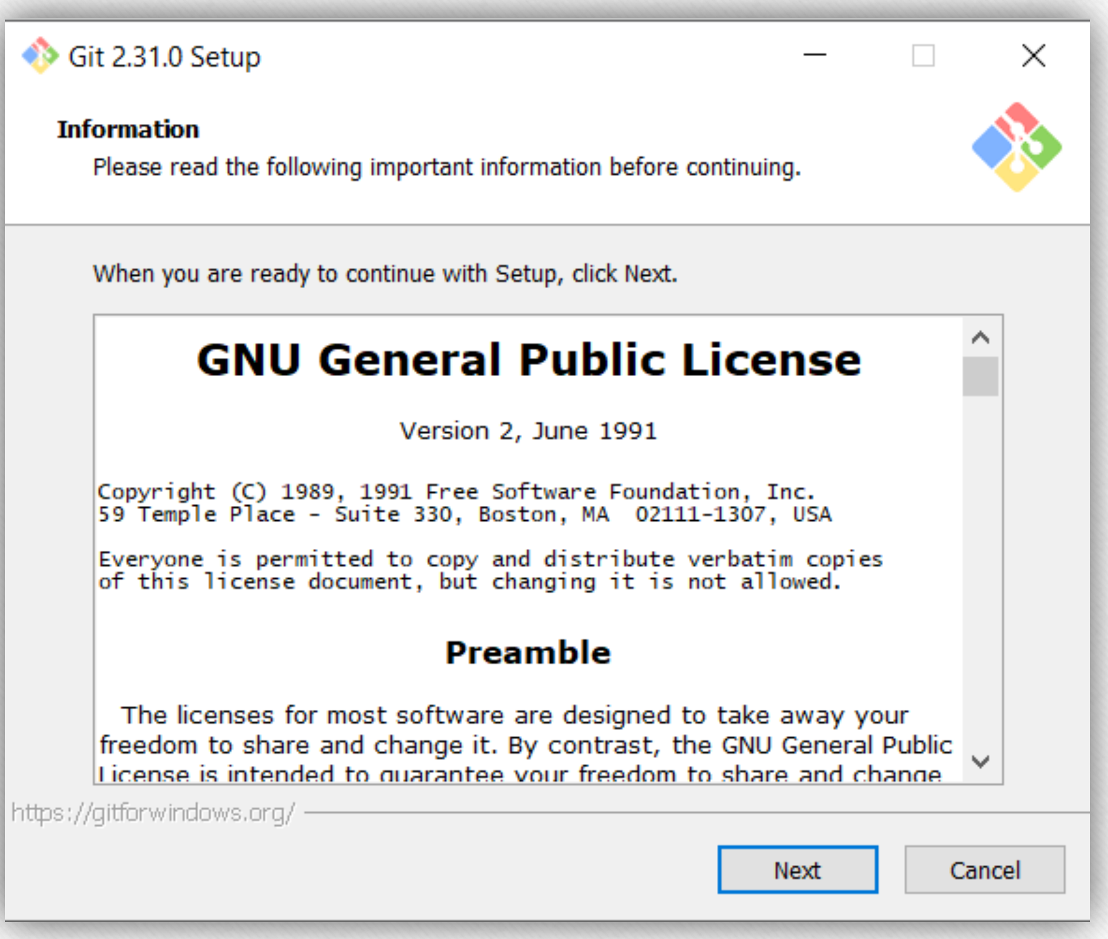
Download Git SCM CMD for windows environment from below link.
www.git-scm.com





TECHLAKE
CLOUD IS THE FUTURE
@trraveendra

Install Git Scm in windows environment. You can go default with all options.


Git 2.31.0 Setup

Information

Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

GNU General Public License

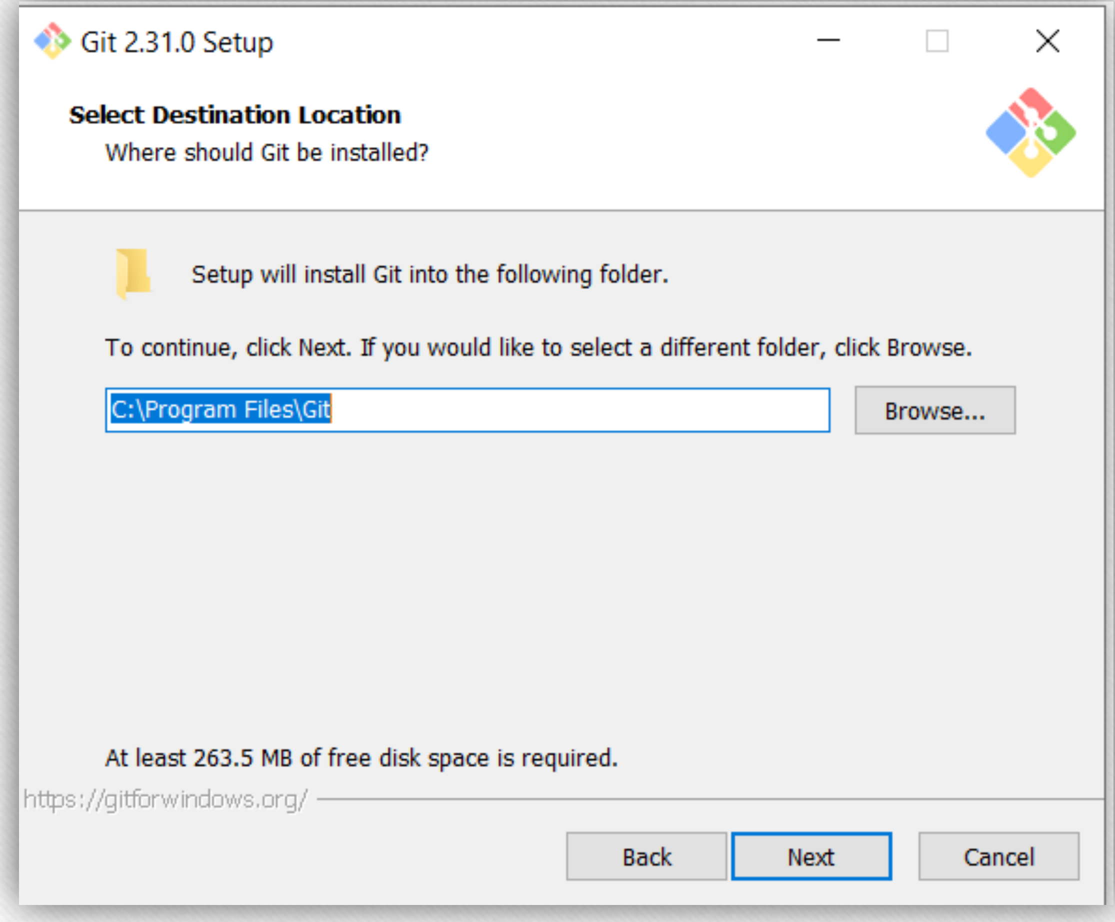
Version 2, June 1991

```
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA  02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
```

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change

https://gitforwindows.org/


Git 2.31.0 Setup

Select Destination Location

Where should Git be installed?

Setup will install Git into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

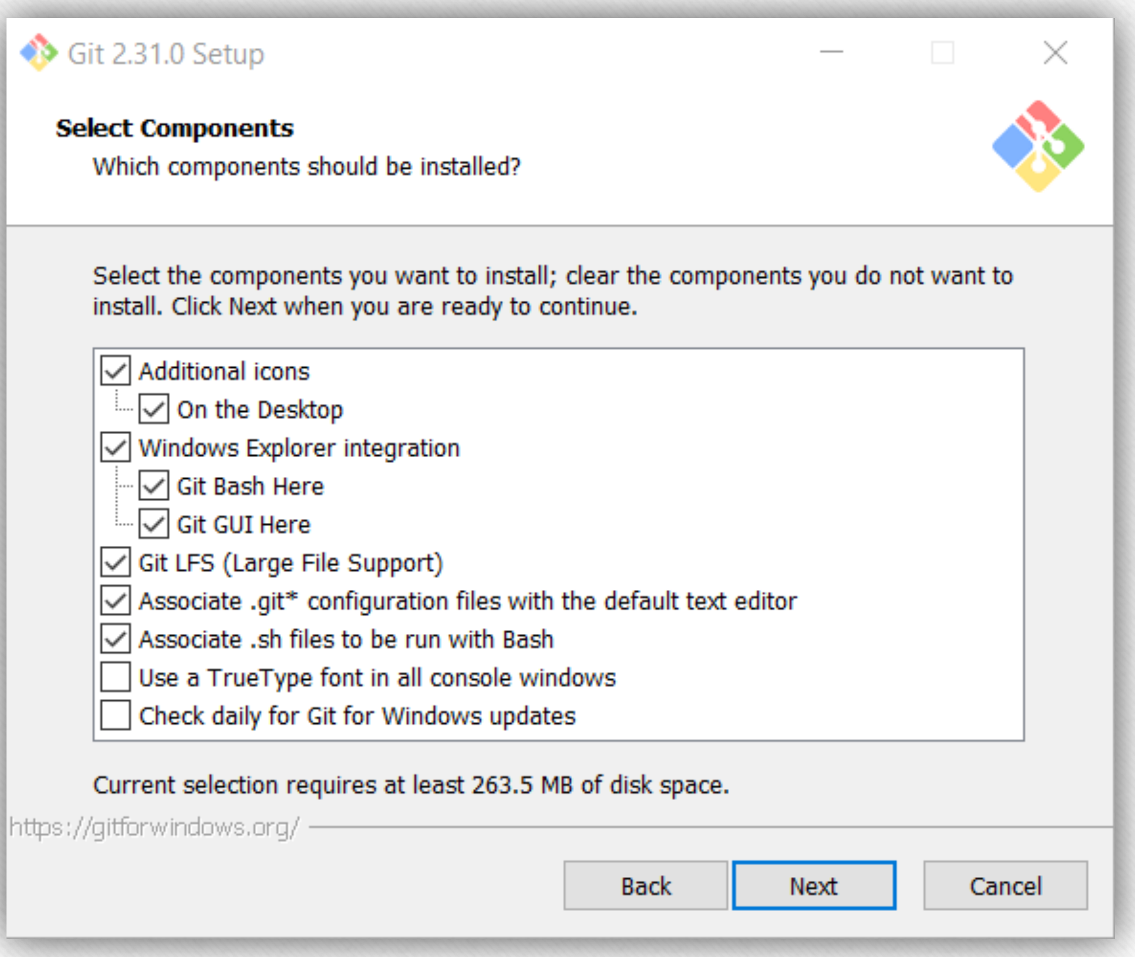
C:\Program Files\Git

At least 263.5 MB of free disk space is required.

https://gitforwindows.org/

TECHLAKE
CLOUD IS THE FUTURE
@trraveendra

Select **On The Desktop** option **Git Bash** and **Git GUI**



Git 2.31.0 Setup

Select Components

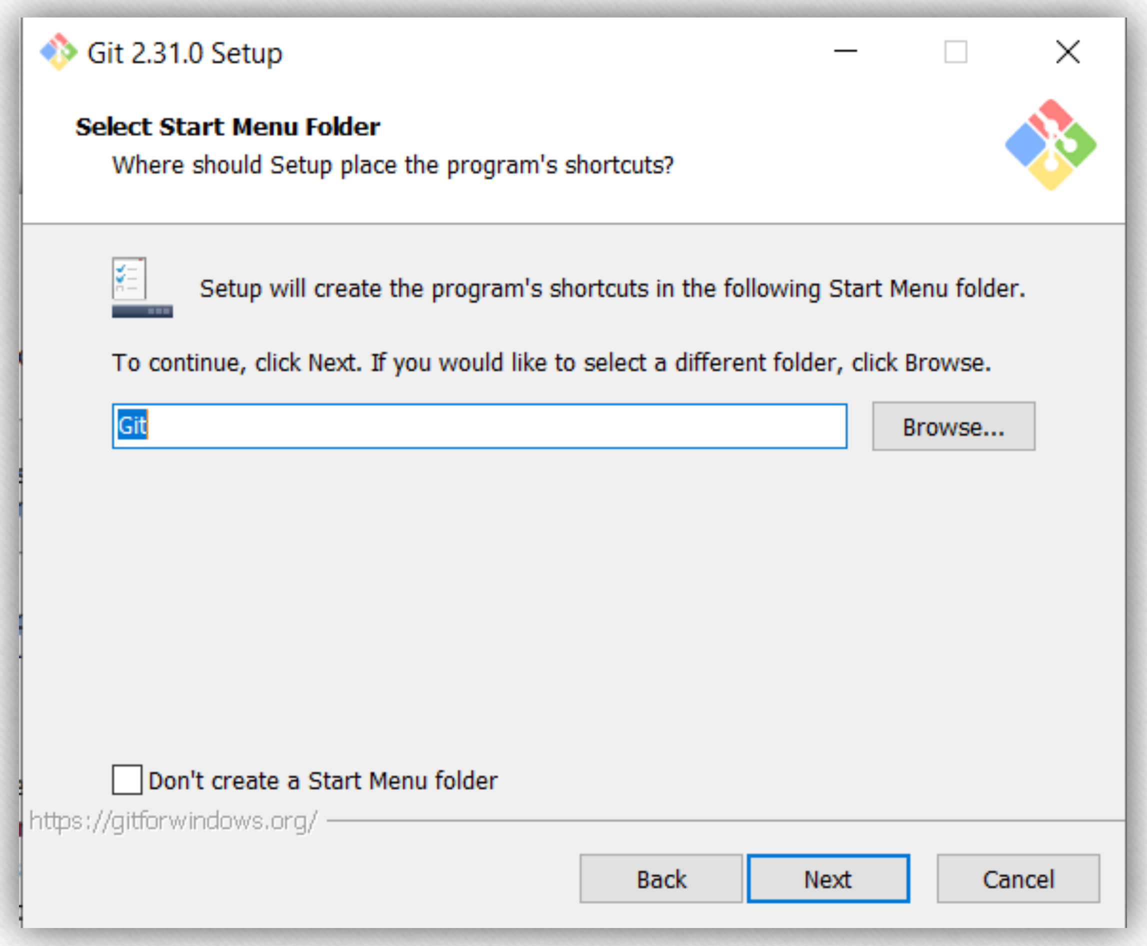
Which components should be installed?

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- [x] Additional icons
  - [x] On the Desktop
- [x] Windows Explorer integration
  - [x] Git Bash Here
  - [x] Git GUI Here
- [x] Git LFS (Large File Support)
- [x] Associate .git* configuration files with the default text editor
- [x] Associate .sh files to be run with Bash
- [ ] Use a TrueType font in all console windows
- [ ] Check daily for Git for Windows updates

Current selection requires at least 263.5 MB of disk space.

https://gitforwindows.org/



Git 2.31.0 Setup

Select Start Menu Folder

Where should Setup place the program's shortcuts?

Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

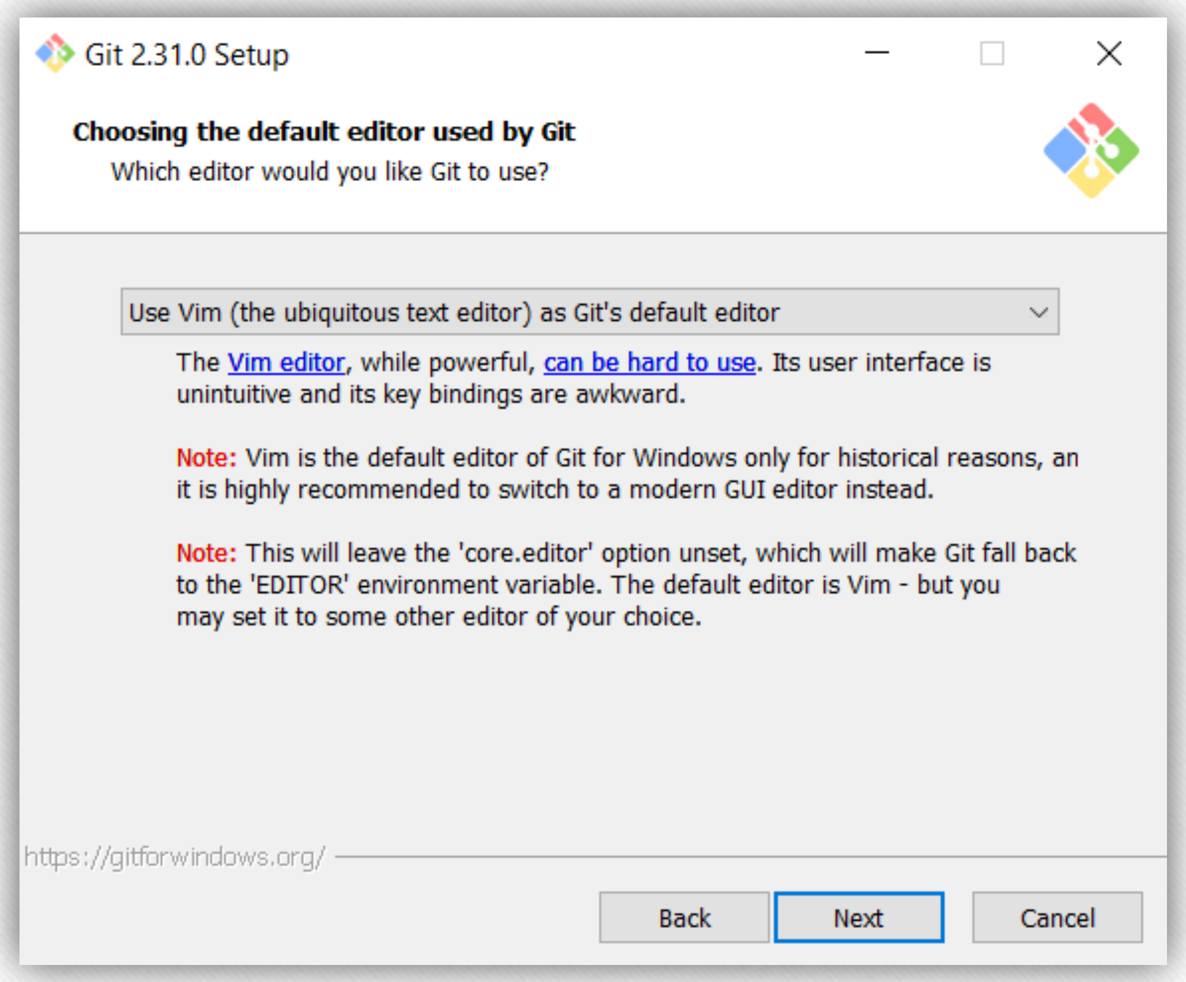
Git

- [ ] Don't create a Start Menu folder

https://gitforwindows.org/

TECHLAKE
CLOUD IS THE FUTURE
@trraveendra

Select **Git From the command line and also from 3rd party software**


Git 2.31.0 Setup

Choosing the default editor used by Git
Which editor would you like Git to use?

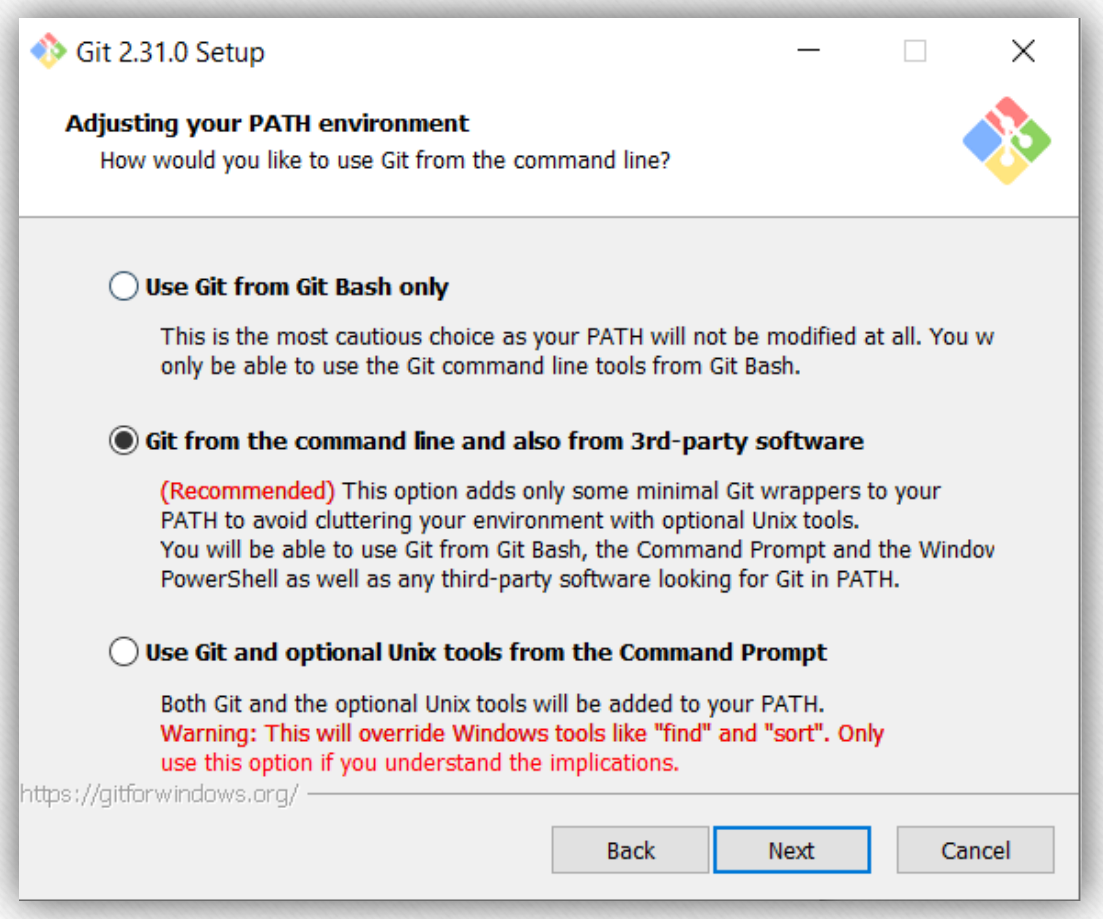
Use Vim (the ubiquitous text editor) as Git's default editor

The Vim editor, while powerful, can be hard to use. Its user interface is unintuitive and its key bindings are awkward.

Note: Vim is the default editor of Git for Windows only for historical reasons, and it is highly recommended to switch to a modern GUI editor instead.

Note: This will leave the 'core.editor' option unset, which will make Git fall back to the 'EDITOR' environment variable. The default editor is Vim - but you may set it to some other editor of your choice.

https://gitforwindows.org/


Git 2.31.0 Setup

Adjusting your PATH environment
How would you like to use Git from the command line?

○ **Use Git from Git Bash only**

This is the most cautious choice as your PATH will not be modified at all. You w
only be able to use the Git command line tools from Git Bash.

● **Git from the command line and also from 3rd-party software**

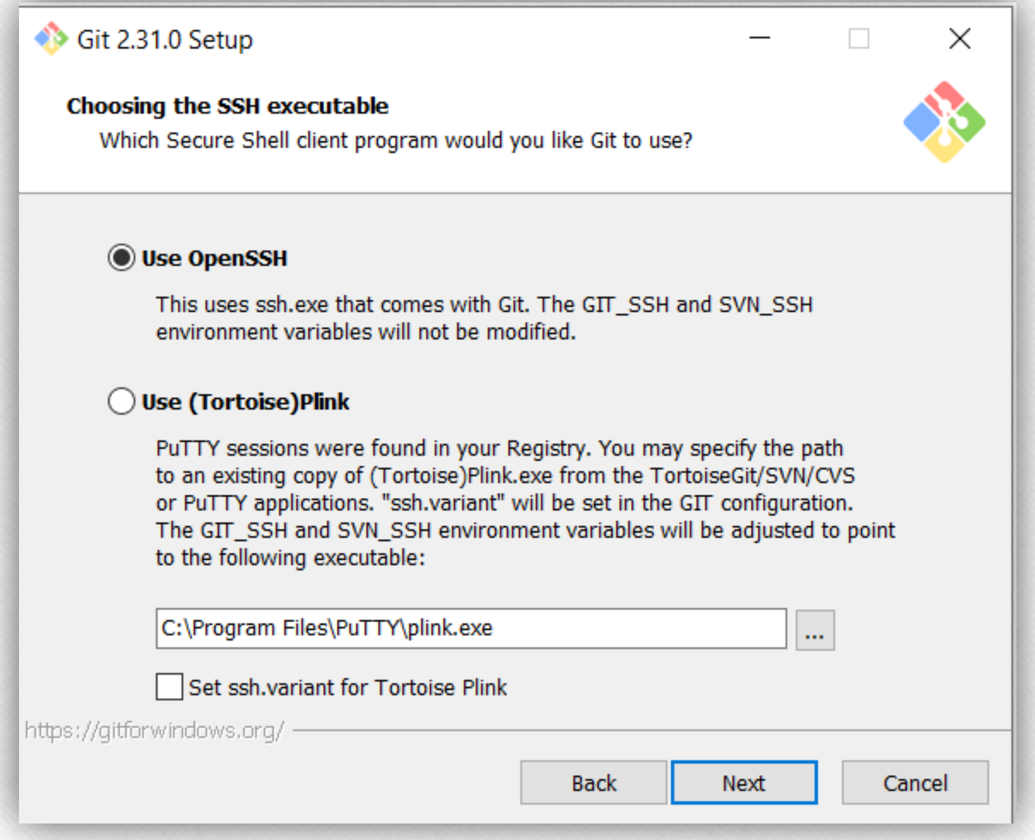
(Recommended) This option adds only some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools.
You will be able to use Git from Git Bash, the Command Prompt and the Windov
PowerShell as well as any third-party software looking for Git in PATH.

○ **Use Git and optional Unix tools from the Command Prompt**

Both Git and the optional Unix tools will be added to your PATH.
Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.

https://gitforwindows.org/

Select **OpenSSH**



Git 2.31.0 Setup

Choosing the SSH executable
Which Secure Shell client program would you like Git to use?

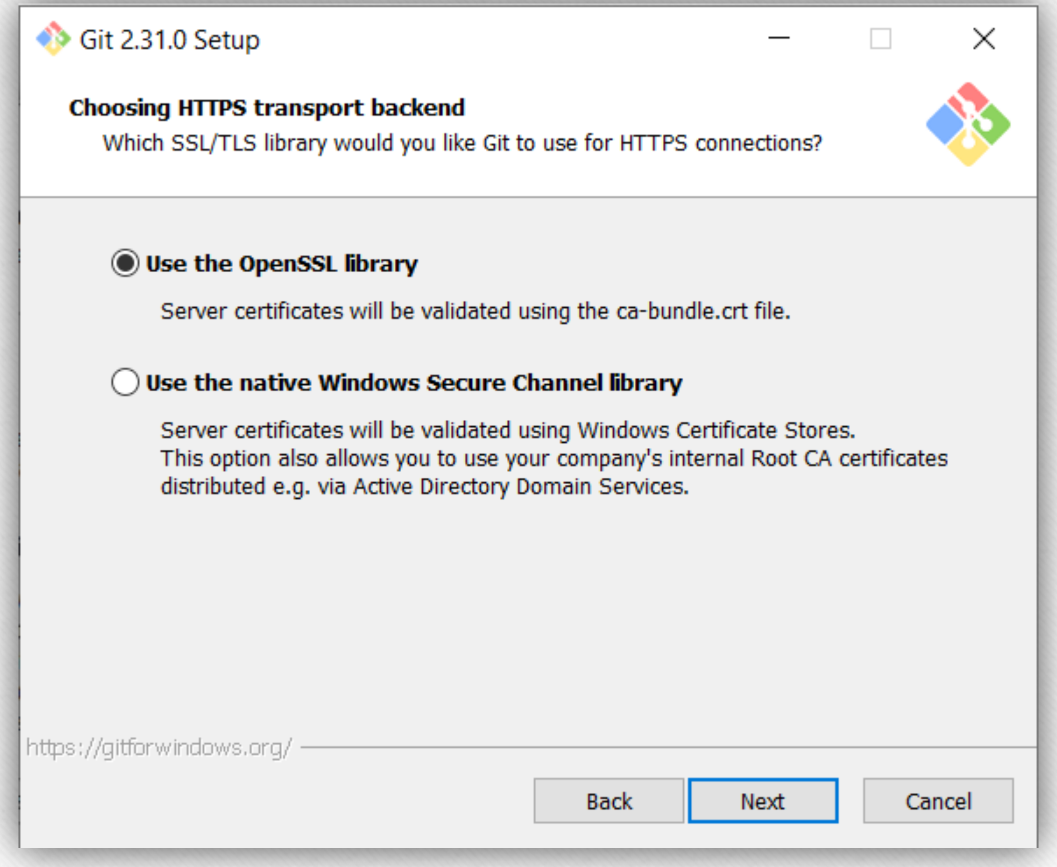
(●) **Use OpenSSH**
This uses ssh.exe that comes with Git. The GIT_SSH and SVN_SSH environment variables will not be modified.

( ) **Use (Tortoise)Plink**
PuTTY sessions were found in your Registry. You may specify the path to an existing copy of (Tortoise)Plink.exe from the TortoiseGit/SVN/CVS or PuTTY applications. "ssh.variant" will be set in the GIT configuration. The GIT_SSH and SVN_SSH environment variables will be adjusted to point to the following executable:

C:\Program Files\PuTTY\plink.exe

☐ Set ssh.variant for Tortoise Plink

https://gitforwindows.org/



Git 2.31.0 Setup

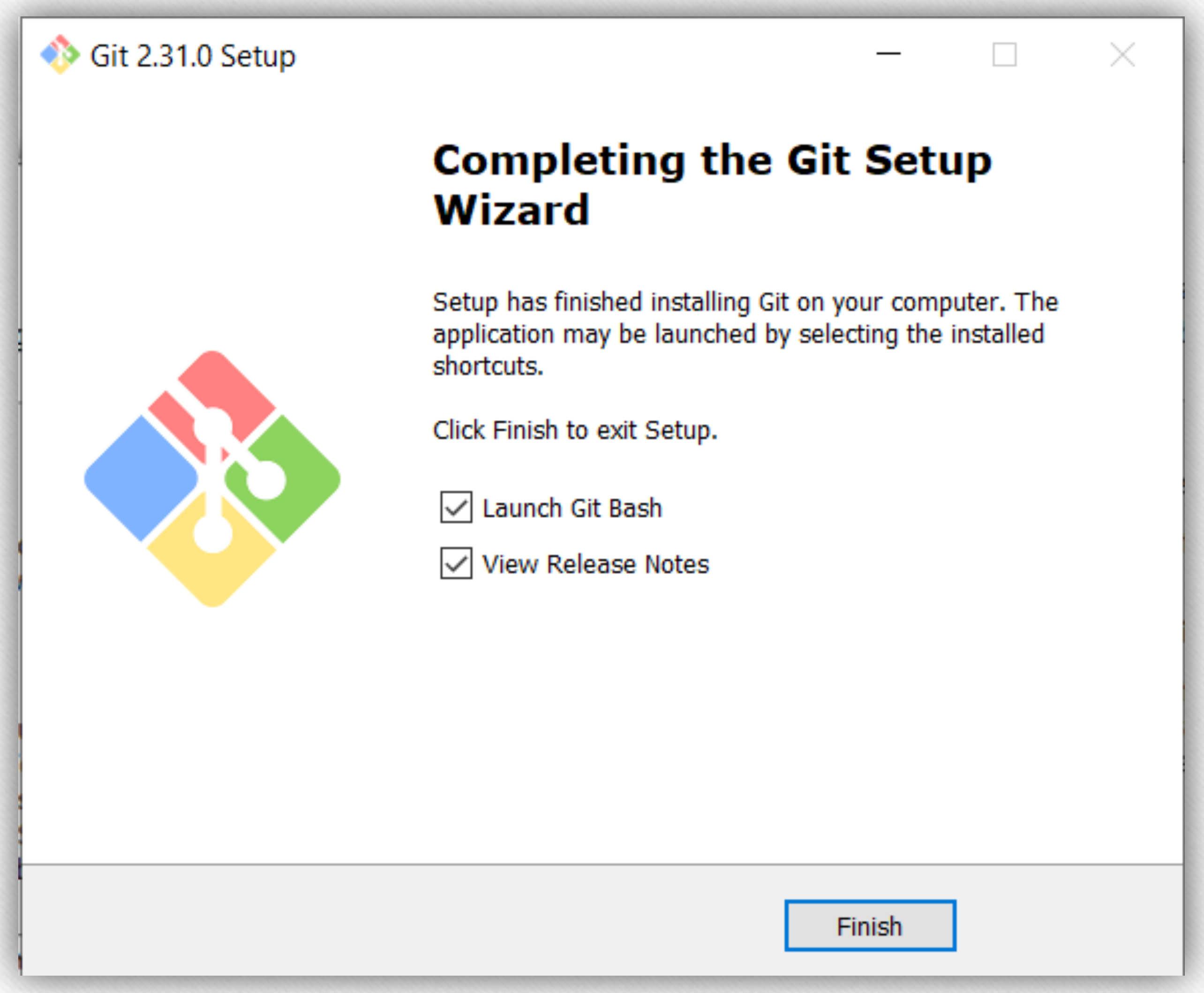
Choosing HTTPS transport backend
Which SSL/TLS library would you like Git to use for HTTPS connections?

(●) **Use the OpenSSL library**
Server certificates will be validated using the ca-bundle.crt file.

( ) **Use the native Windows Secure Channel library**
Server certificates will be validated using Windows Certificate Stores.
This option also allows you to use your company's internal Root CA certificates distributed e.g. via Active Directory Domain Services.

https://gitforwindows.org/

Select **Launch Git Bash** it will open **Git Bash**



Completing the Git Setup Wizard

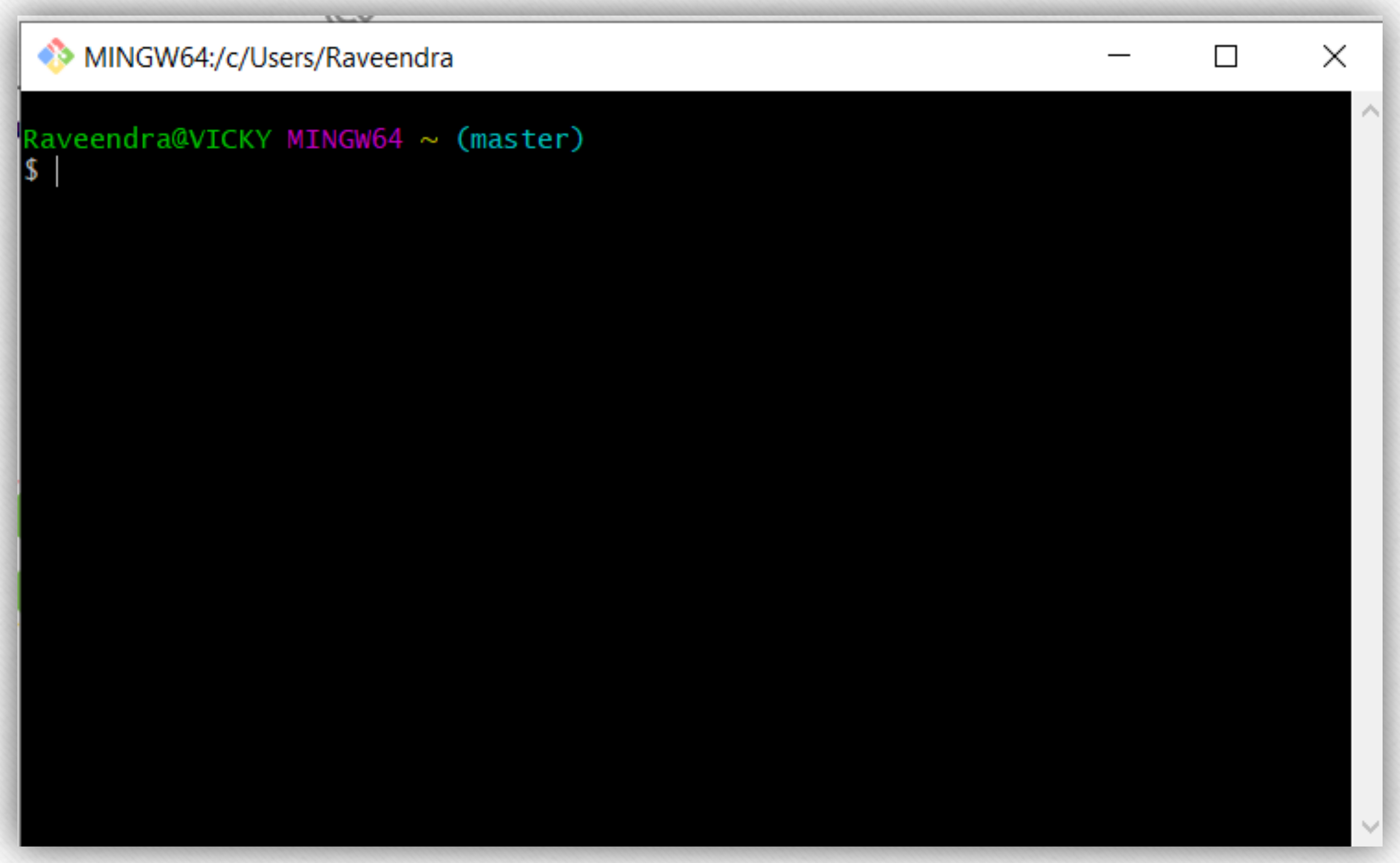
Setup has finished installing Git on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

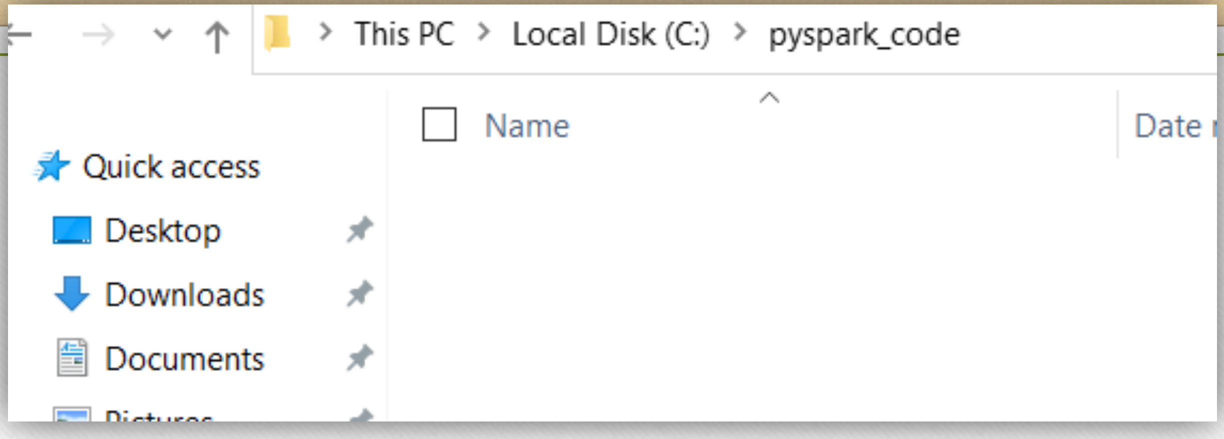
- [✓] Launch Git Bash
- [✓] View Release Notes

TECHLAKE
CLOUD IS THE FUTURE
@trraveendra

Git Bash : here we can use all GIT Commands for
1) cloning repositories
2) Adding new code changes to repository
3) Committing code changes
4) Pushing committed changes to remote repository

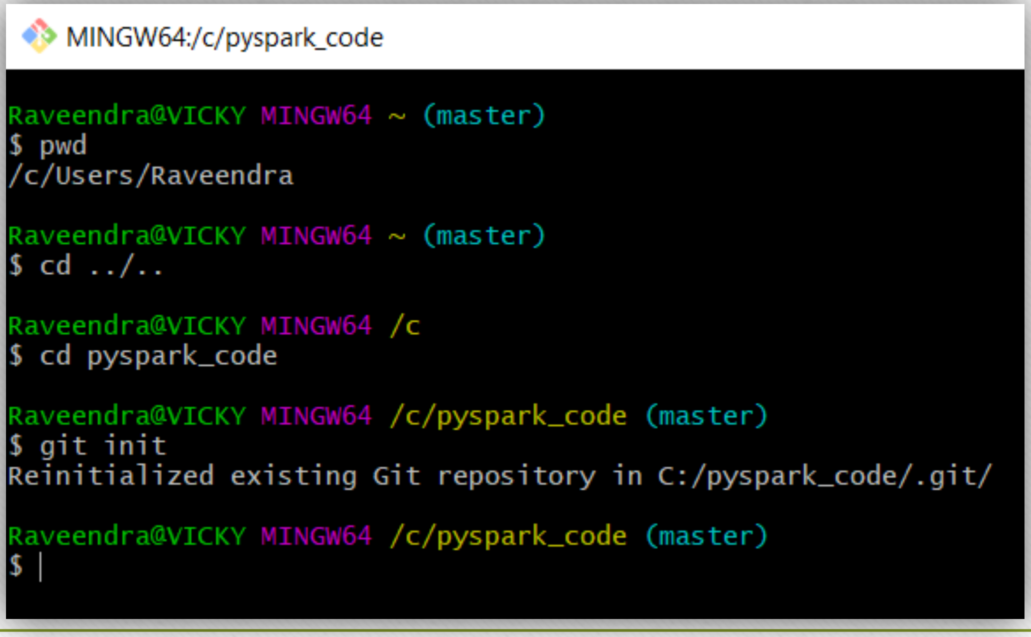

MINGW64:/c/Users/Raveendra

```
Raveendra@VICKY MINGW64 ~ (master)
$ |
```



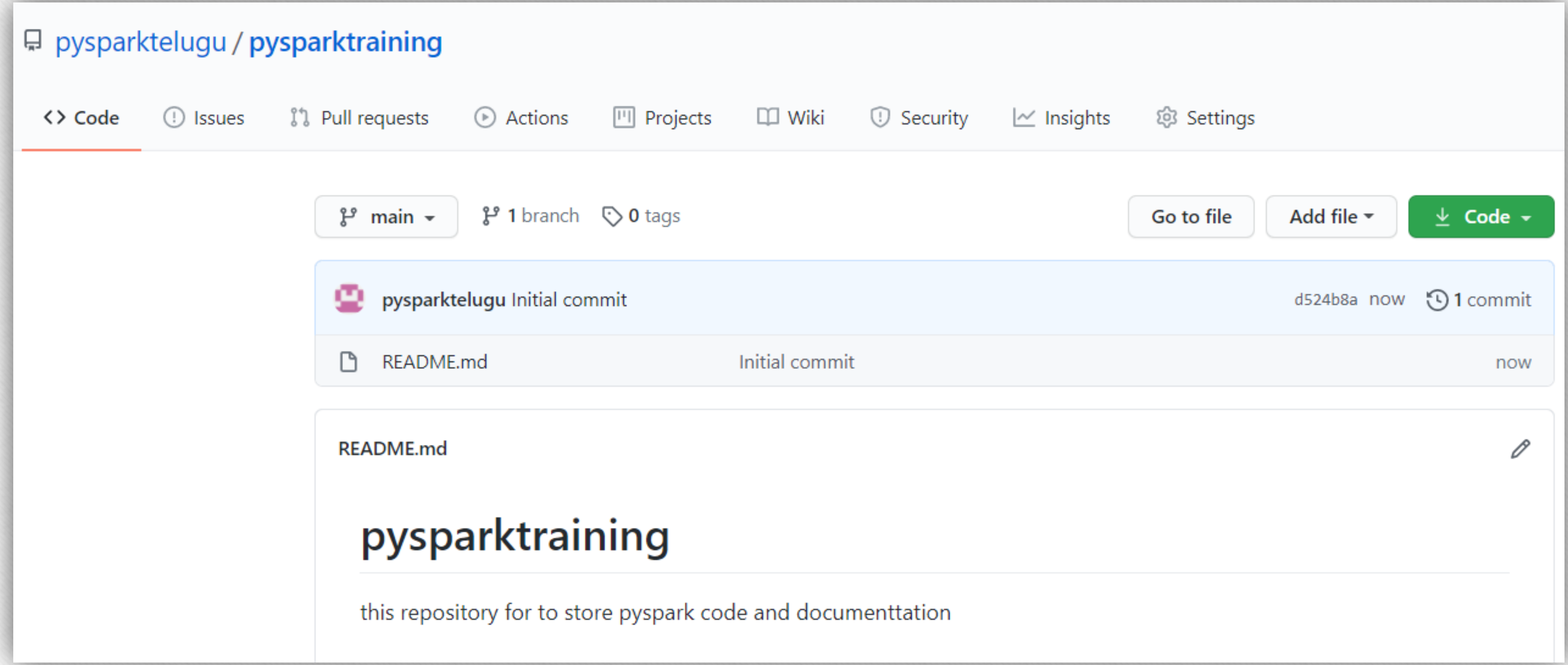
Getting Current Working directory using **pwd**
Changing directory using **cd**

init : initializing repository to particular directory.



```
MINGW64:/c/pyspark_code

Raveendra@VICKY MINGW64 ~ (master)
$ pwd
/c/Users/Raveendra

Raveendra@VICKY MINGW64 ~ (master)
$ cd ../..

Raveendra@VICKY MINGW64 /c
$ cd pyspark_code

Raveendra@VICKY MINGW64 /c/pyspark_code (master)
$ git init
Reinitialized existing Git repository in C:/pyspark_code/.git/

Raveendra@VICKY MINGW64 /c/pyspark_code (master)
$ |
```

Cloning github repository



pysparktelugu Initial commit d524b8a now 1 commit

README.md Initial commit now

README.md

pysparktraining

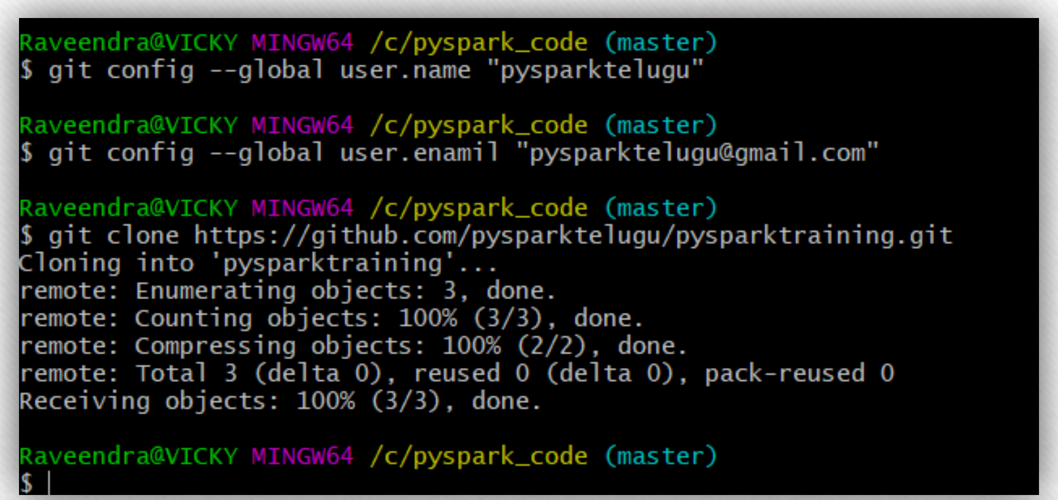
this repository for to store pyspark code and documenttation

TECHLAKE
CLOUD IS THE FUTURE
@trraveendra

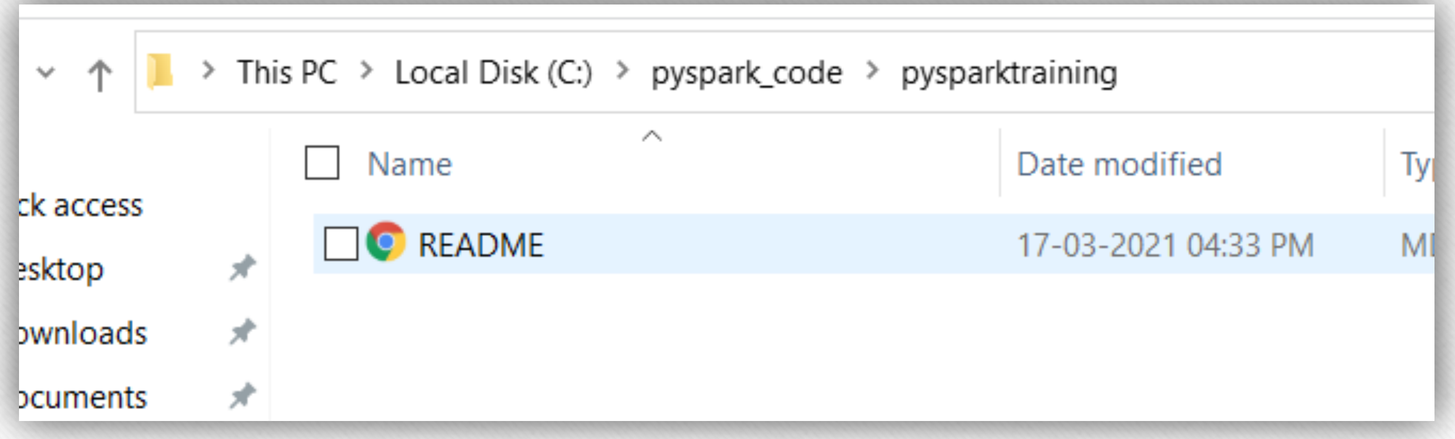
After installing Git, the first thing to do is to set your **name** and **email**

git config --global user.name “pysparktelugu"
git config --global user.email “pysparktelugu@gmail.com"

git clone **giturl - The second image is showing cloned repository from github.com**



```
Raveendra@VICKY MINGW64 /c/pyspark_code (master)
$ git config --global user.name "pysparktelugu"

Raveendra@VICKY MINGW64 /c/pyspark_code (master)
$ git config --global user.enamil "pysparktelugu@gmail.com"

Raveendra@VICKY MINGW64 /c/pyspark_code (master)
$ git clone https://github.com/pysparktelugu/pysparktraining.git
Cloning into 'pysparktraining'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

Raveendra@VICKY MINGW64 /c/pyspark_code (master)
$ |
```

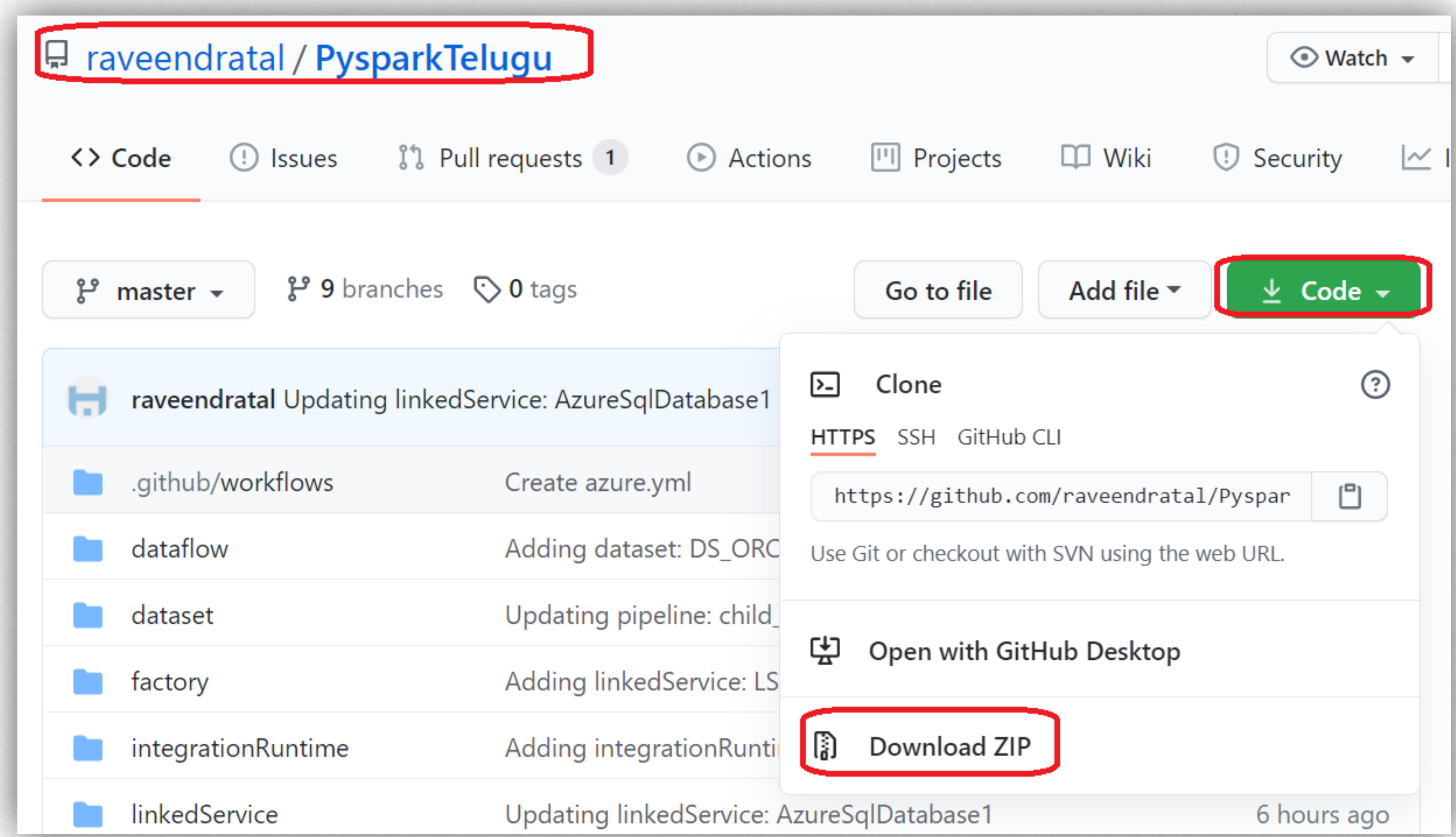


This PC > Local Disk (C:) > pyspark_code > pysparktraining

Name	Date modified	Ty
README	17-03-2021 04:33 PM	MI

Now we will download entire repository as a zip file and extract in local system which is cloned directory in previous step

Open github url https://github.com/raveendratal/PysparkTelugu   Download code in zip file.



raveendratal Updating linkedService: AzureSqlDatabase1

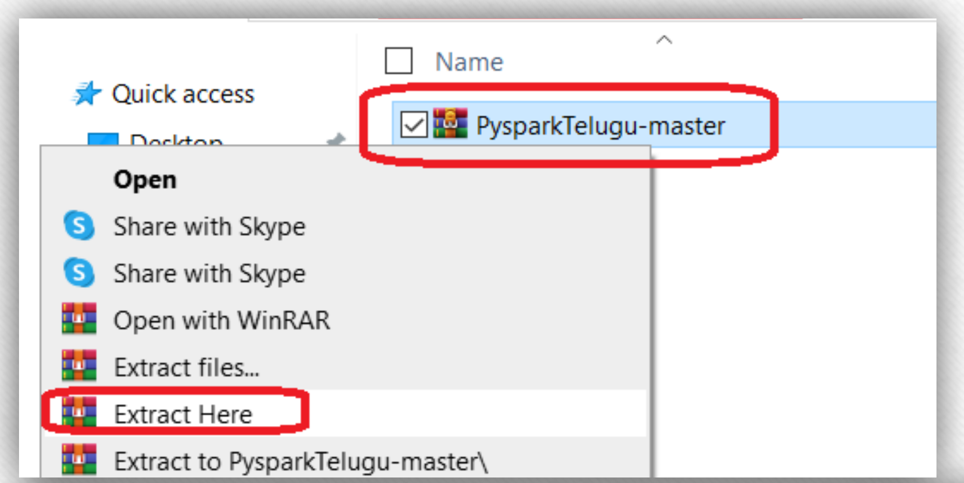
https://github.com/raveendratal/Pyspar

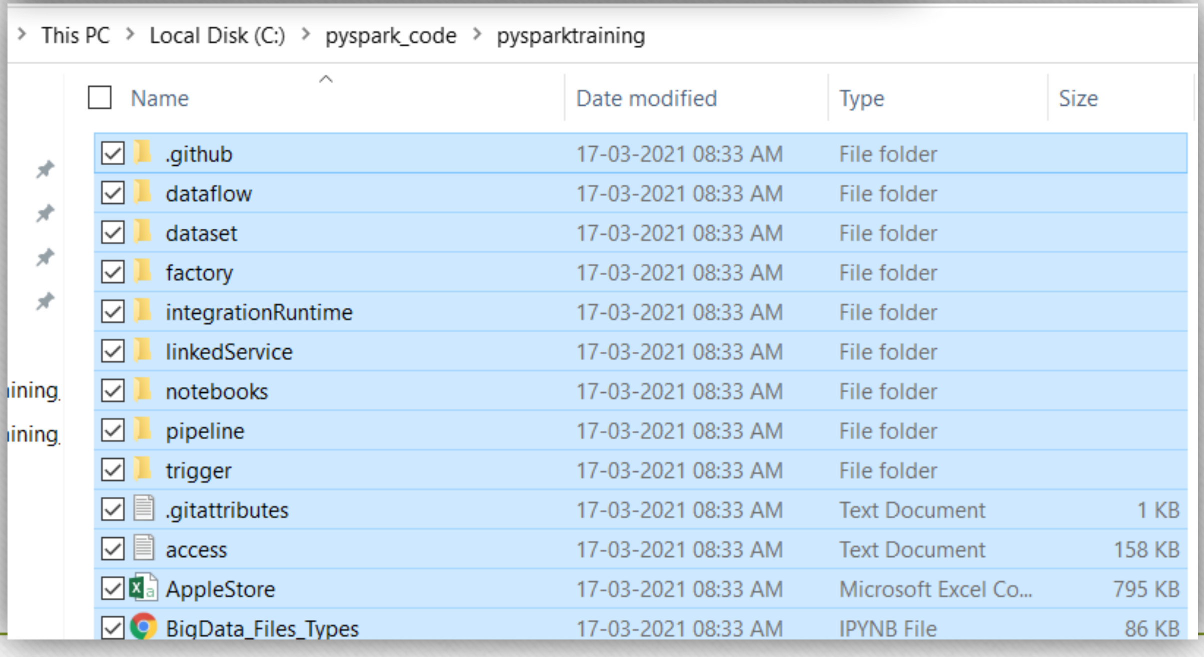
Use Git or checkout with SVN using the web URL.

.github/workflows	Create azure.yml	
dataflow	Adding dataset: DS_ORC	
dataset	Updating pipeline: child_	
factory	Adding linkedService: LS	
integrationRuntime	Adding integrationRunti	
linkedService	Updating linkedService: AzureSqlDatabase1	6 hours ago

TECHLAKE
CLOUD IS THE FUTURE
@trraveendra

extract downloaded pysparktelugu-master zip file into C:\pyspark_code\pysparktraining. After Extraction delete zip file. Any way its not required.

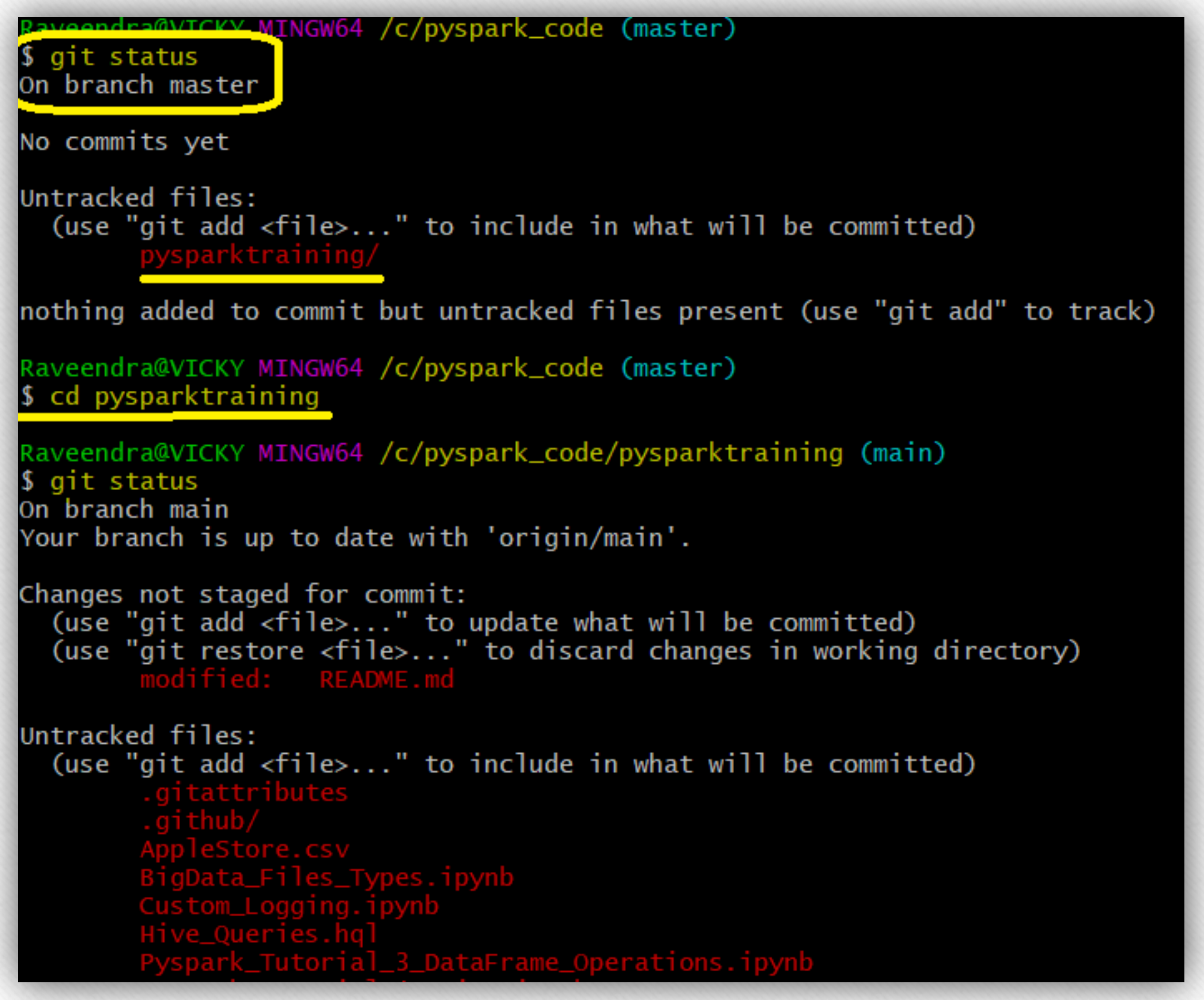



This PC › Local Disk (C:) › pyspark_code › pysparktraining

Name	Date modified	Type	Size
.github	17-03-2021 08:33 AM	File folder	
dataflow	17-03-2021 08:33 AM	File folder	
dataset	17-03-2021 08:33 AM	File folder	
factory	17-03-2021 08:33 AM	File folder	
integrationRuntime	17-03-2021 08:33 AM	File folder	
linkedService	17-03-2021 08:33 AM	File folder	
notebooks	17-03-2021 08:33 AM	File folder	
pipeline	17-03-2021 08:33 AM	File folder	
trigger	17-03-2021 08:33 AM	File folder	
.gitattributes	17-03-2021 08:33 AM	Text Document	1 KB
access	17-03-2021 08:33 AM	Text Document	158 KB
AppleStore	17-03-2021 08:33 AM	Microsoft Excel Co...	795 KB
BigData_Files_Types	17-03-2021 08:33 AM	IPYNB File	86 KB

TECHLAKE
CLOUD IS THE FUTURE.
@trraveendra

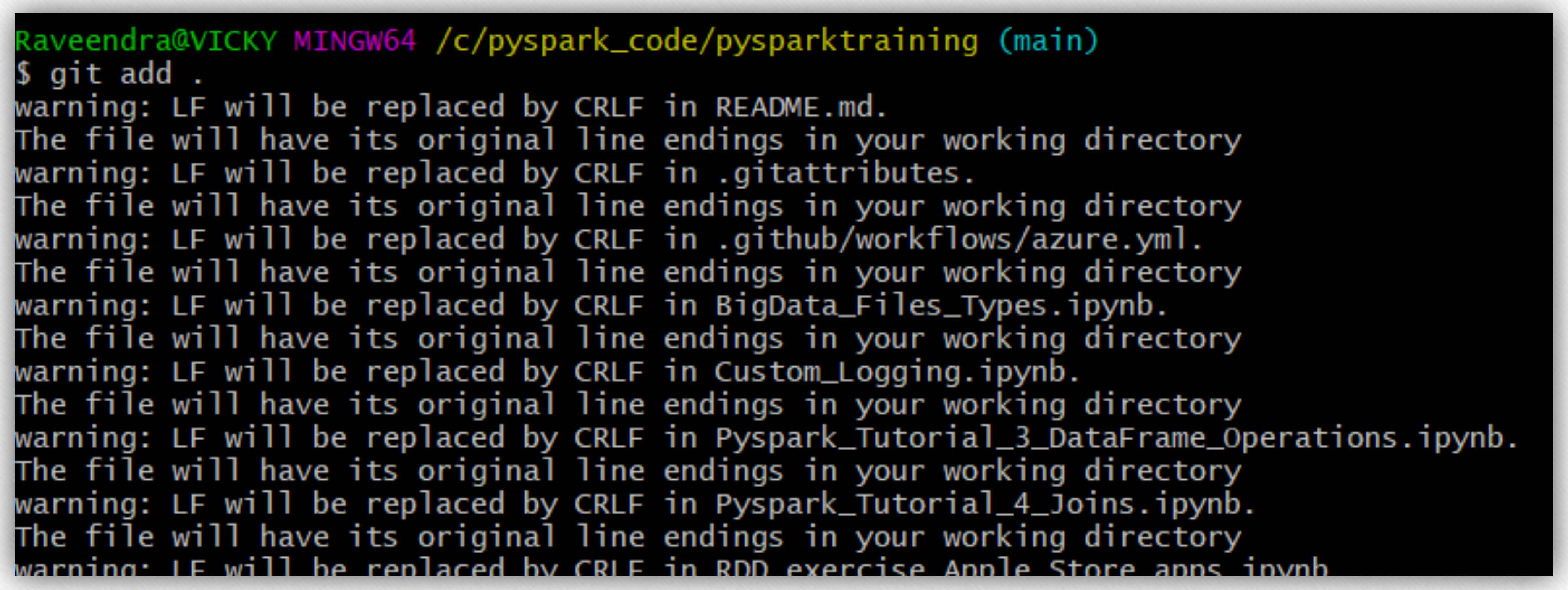
Git status : git status will show the any new changes in repository



```
Raveendra@VICKY MINGW64 /c/pyspark_code (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        pysparktraining/

nothing added to commit but untracked files present (use "git add" to track)

Raveendra@VICKY MINGW64 /c/pyspark_code (master)
$ cd pysparktraining

Raveendra@VICKY MINGW64 /c/pyspark_code/pysparktraining (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitattributes
        .github/
        AppleStore.csv
        BigData_Files_Types.ipynb
        Custom_Logging.ipynb
        Hive_Queries.hql
        Pyspark_Tutorial_3_DataFrame_Operations.ipynb
```

TECHLAKE
CLOUD IS THE FUTURE
@trraveendra

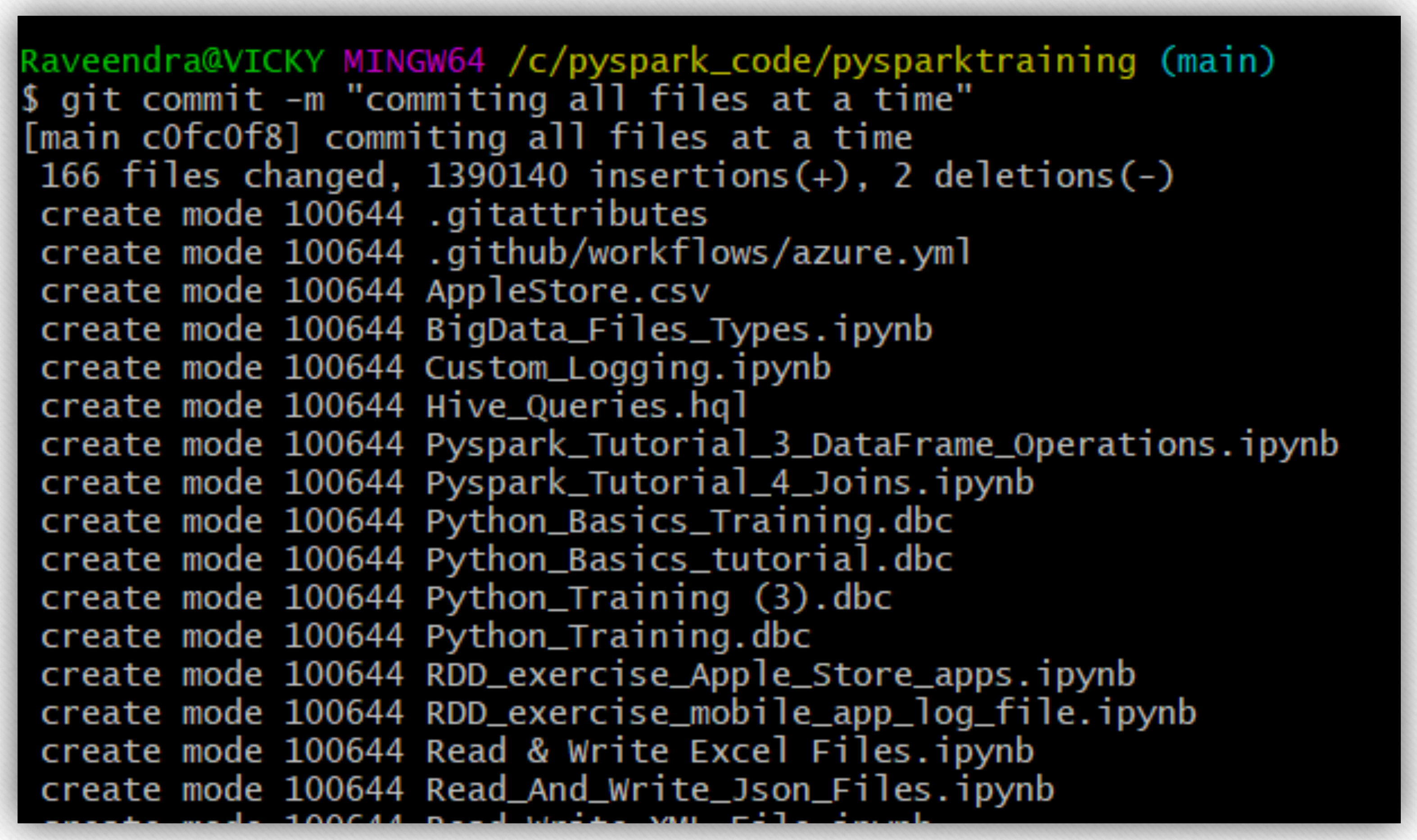
Using git add adding all newly added files into staging area. git add . (. Means it will consider all files)



```
Raveendra@VICKY MINGW64 /c/pyspark_code/pysparktraining (main)
$ git add .
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in .gitattributes.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in .github/workflows/azure.yml.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in BigData_Files_Types.ipynb.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in Custom_Logging.ipynb.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in Pyspark_Tutorial_3_DataFrame_Operations.ipynb.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in Pyspark_Tutorial_4_Joins.ipynb.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in RDD exercise Apple Store apps.ipynb
```

TECHLAKE
CLOUD IS THE FUTURE.
@trraveendra

Git commit : using **git commit** we can commit all new changes from staging to local repository
-M " comments" - m option we can use for providing commit comments.

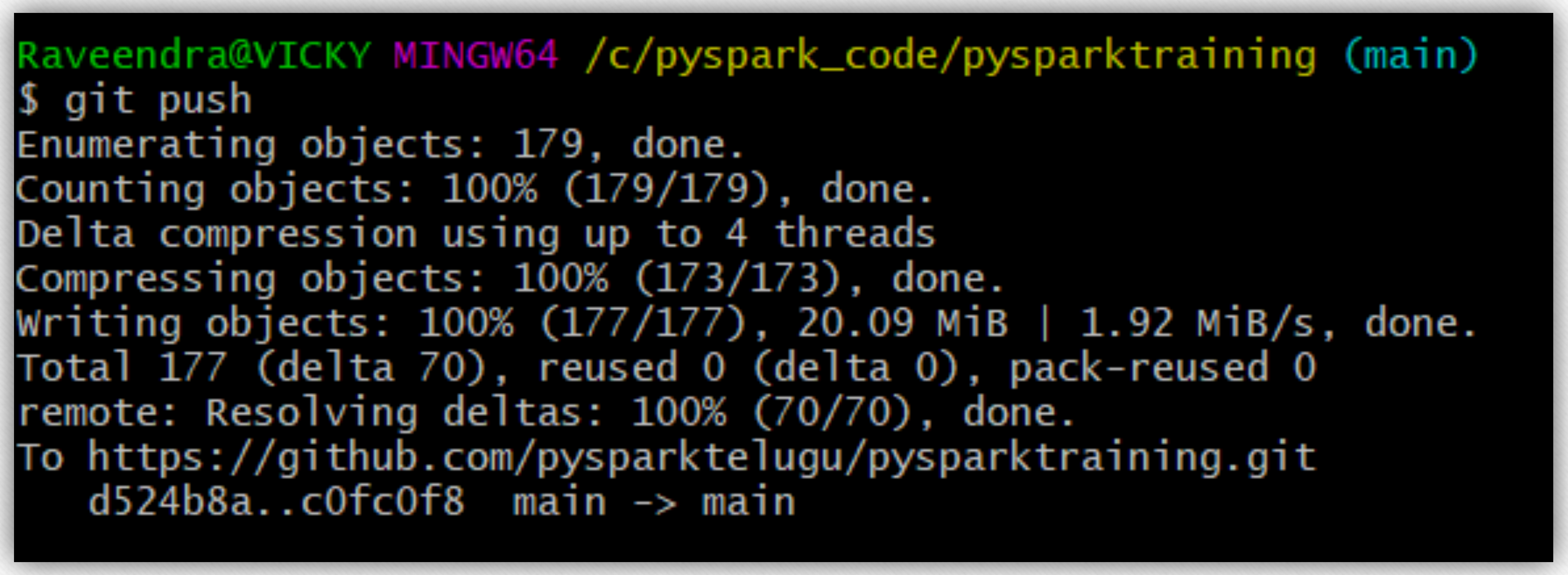


```
Raveendra@VICKY MINGW64 /c/pyspark_code/pysparktraining (main)
$ git commit -m "commiting all files at a time"
[main c0fc0f8] commiting all files at a time
 166 files changed, 1390140 insertions(+), 2 deletions(-)
 create mode 100644 .gitattributes
 create mode 100644 .github/workflows/azure.yml
 create mode 100644 AppleStore.csv
 create mode 100644 BigData_Files_Types.ipynb
 create mode 100644 Custom_Logging.ipynb
 create mode 100644 Hive_Queries.hql
 create mode 100644 Pyspark_Tutorial_3_DataFrame_Operations.ipynb
 create mode 100644 Pyspark_Tutorial_4_Joins.ipynb
 create mode 100644 Python_Basics_Training.dbc
 create mode 100644 Python_Basics_tutorial.dbc
 create mode 100644 Python_Training (3).dbc
 create mode 100644 Python_Training.dbc
 create mode 100644 RDD_exercise_Apple_Store_apps.ipynb
 create mode 100644 RDD_exercise_mobile_app_log_file.ipynb
 create mode 100644 Read & Write Excel Files.ipynb
 create mode 100644 Read_And_Write_Json_Files.ipynb
```

TECHLAKE
CLOUD IS THE FUTURE.
@trraveendra

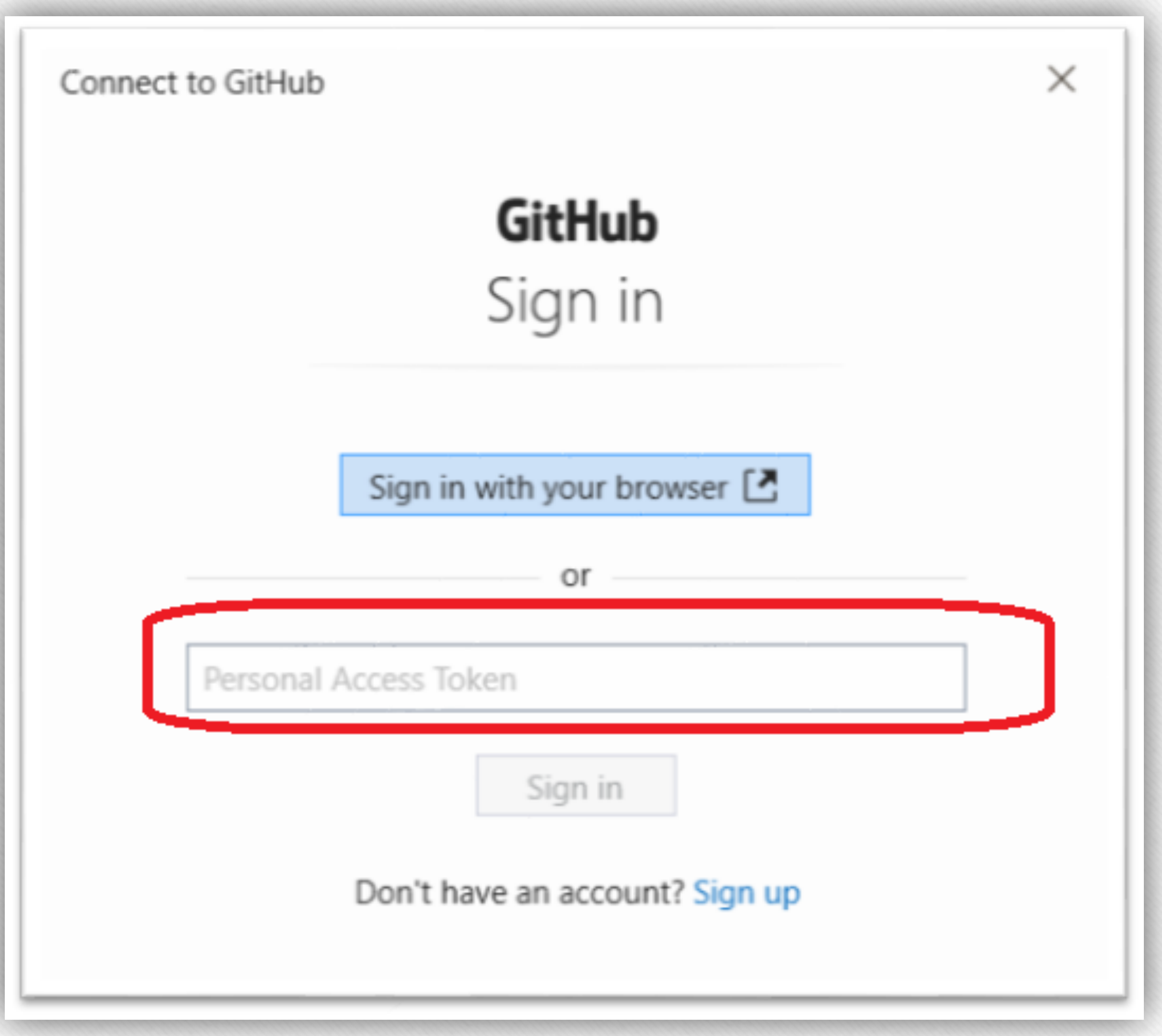
Git push : using **git push** command we can push newly committed changes from **local repository** to **Remote repository**.

First time while using **git push** command it will ask authentication. We can give github.com **access token** or **user-name and password**



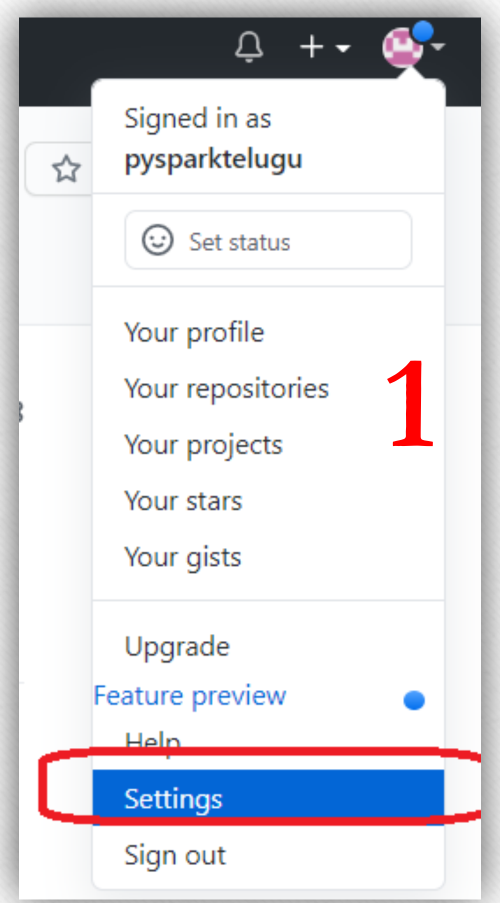
```
Raveendra@VICKY MINGW64 /c/pyspark_code/pysparktraining (main)
$ git push
Enumerating objects: 179, done.
Counting objects: 100% (179/179), done.
Delta compression using up to 4 threads
Compressing objects: 100% (173/173), done.
Writing objects: 100% (177/177), 20.09 MiB | 1.92 MiB/s, done.
Total 177 (delta 70), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (70/70), done.
To https://github.com/pysparktelugu/pysparktraining.git
   d524b8a..c0fc0f8  main -> main
```

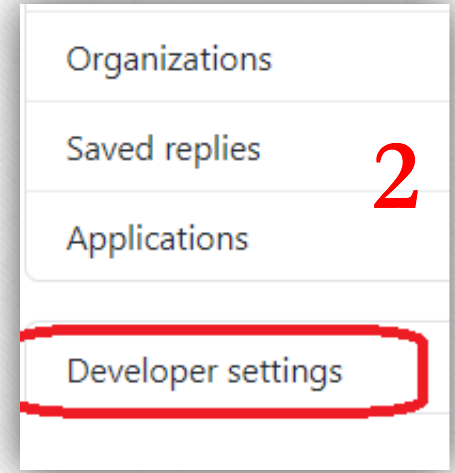
It will open a window like below. You can choose any of the option. Next slide I have given steps for creating **access token**

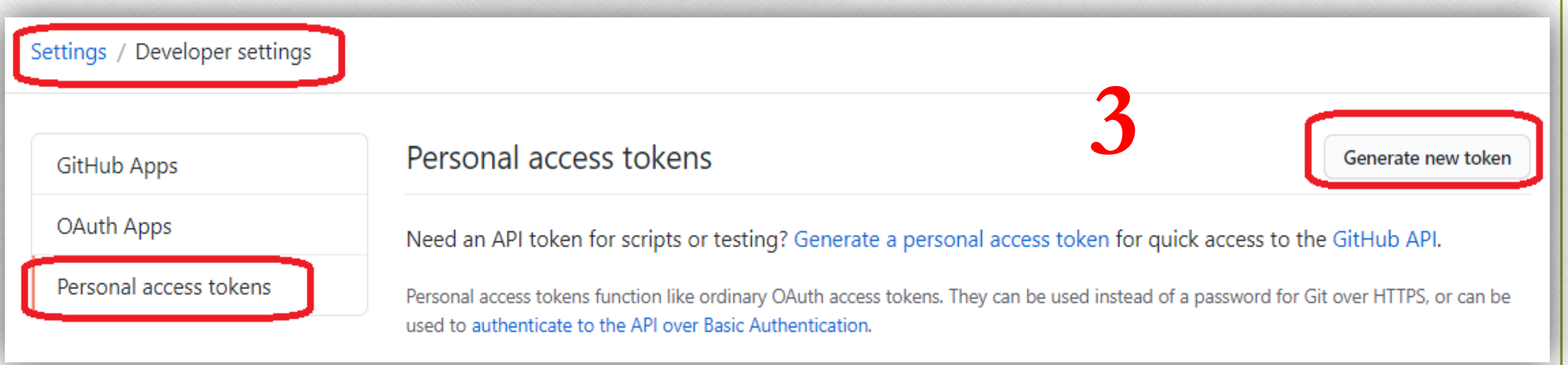


TECHLAKE
CLOUD IS THE FUTURE.
@trraveendra

Login into www.github.com account and go to => profile => settings =>
Left Side => Developer Settings = > Personal Access Tokens.
Click on **Generate New Token**



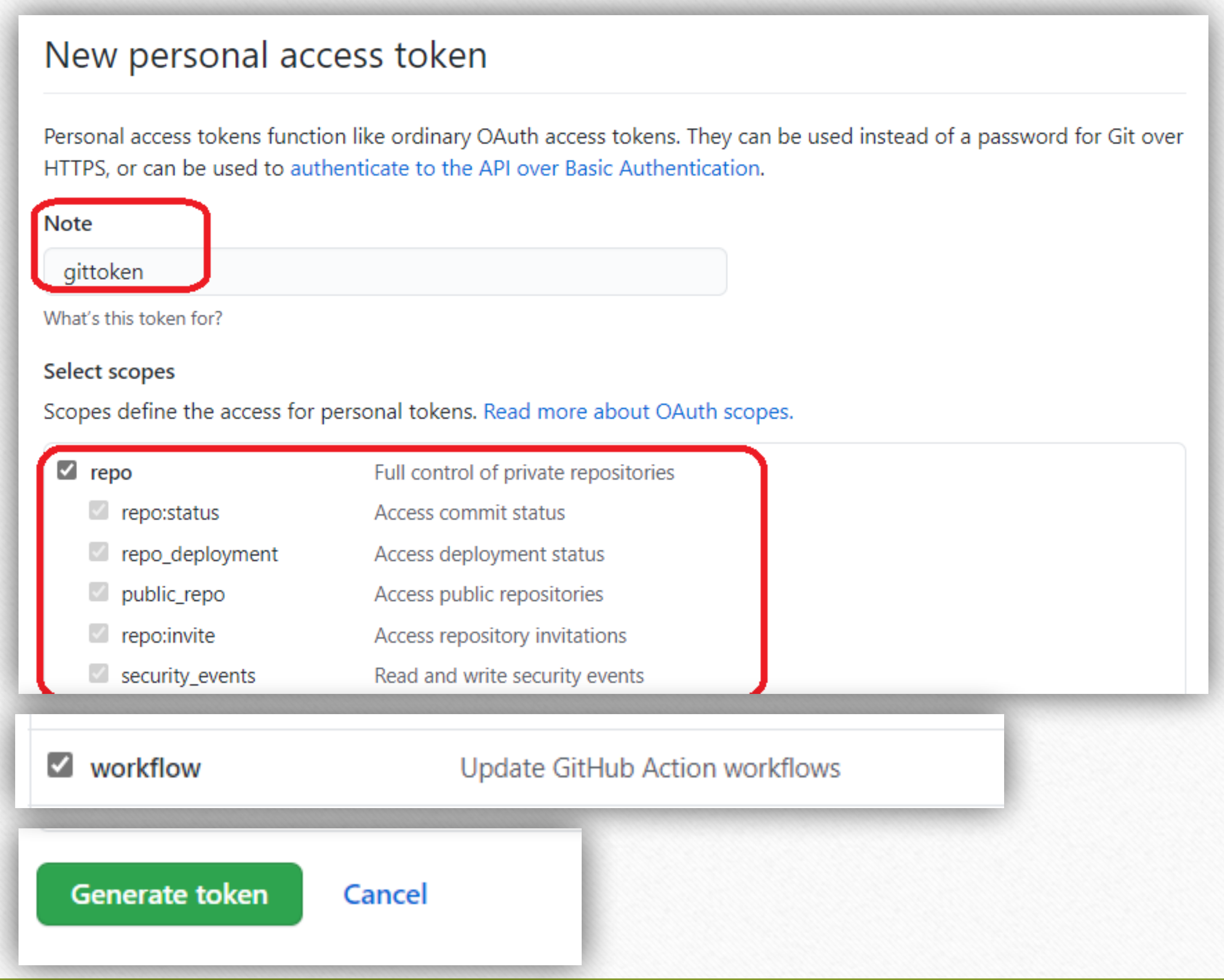




Need an API token for scripts or testing? Generate a personal access token for quick access to the GitHub API.

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Enter name and select **scopes** as listed below. Repo options and **workflow** is mandatory



New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Note

gittoken

What's this token for?

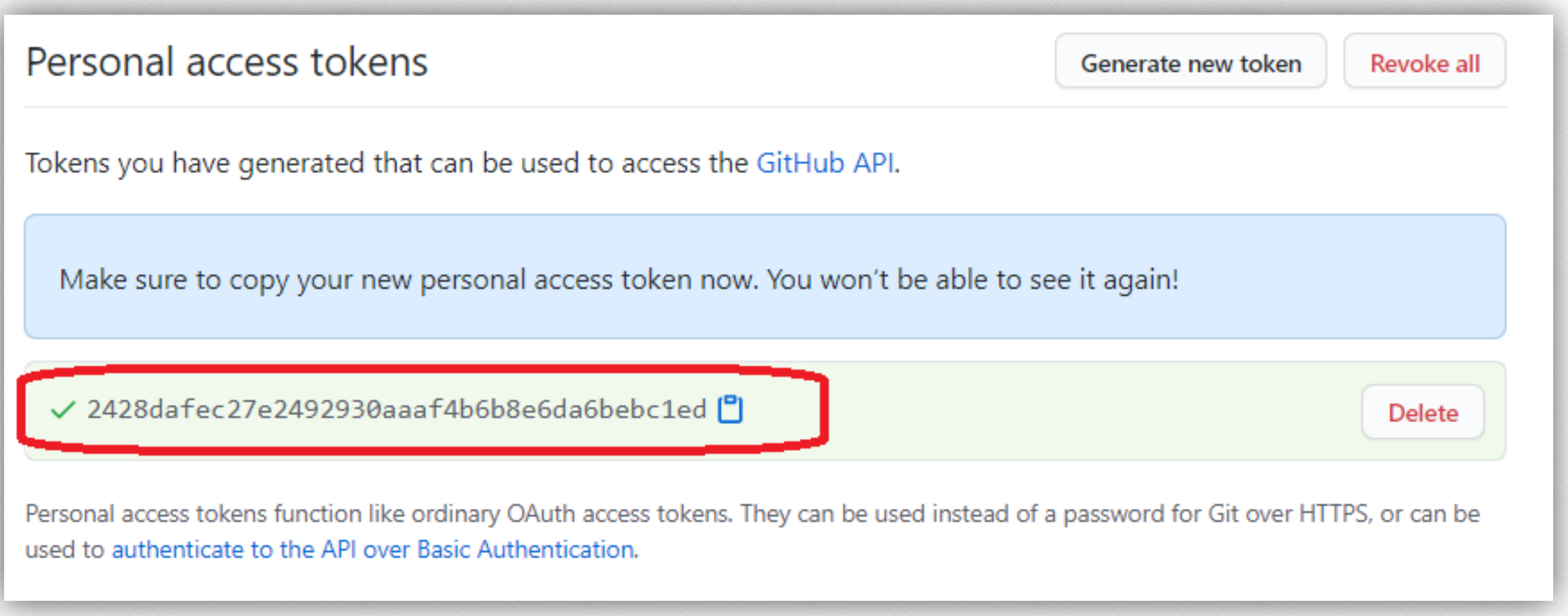
Select scopes

Scopes define the access for personal tokens. Read more about OAuth scopes.

☑ repo	Full control of private repositories
☑ repo:status	Access commit status
☑ repo_deployment	Access deployment status
☑ public_repo	Access public repositories
☑ repo:invite	Access repository invitations
☑ security_events	Read and write security events
☑ workflow	Update GitHub Action workflows

TECHLAKE
CLOUD IS THE FUTURE
@trraveendra

Access token is generated. Copy that and provide in authentication window which is asking while pushing the new Changes to remote repository.



Personal access tokens

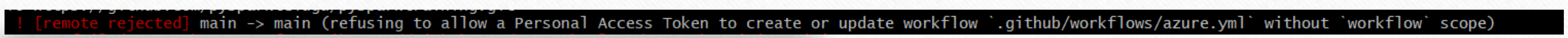
Tokens you have generated that can be used to access the GitHub API.

Make sure to copy your new personal access token now. You won't be able to see it again!

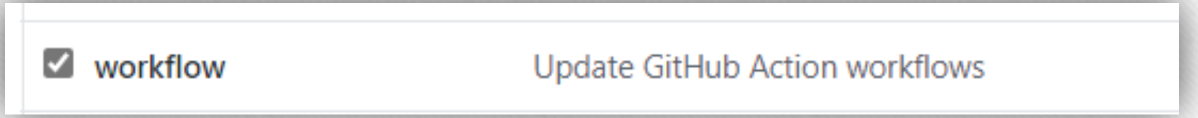
✓ 2428dafec27e2492930aaaf4b6b8e6da6bebc1ed

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Note: If you are not selecting **workflow** option while generating token. It will throw error while moving new changes to Remote repository.

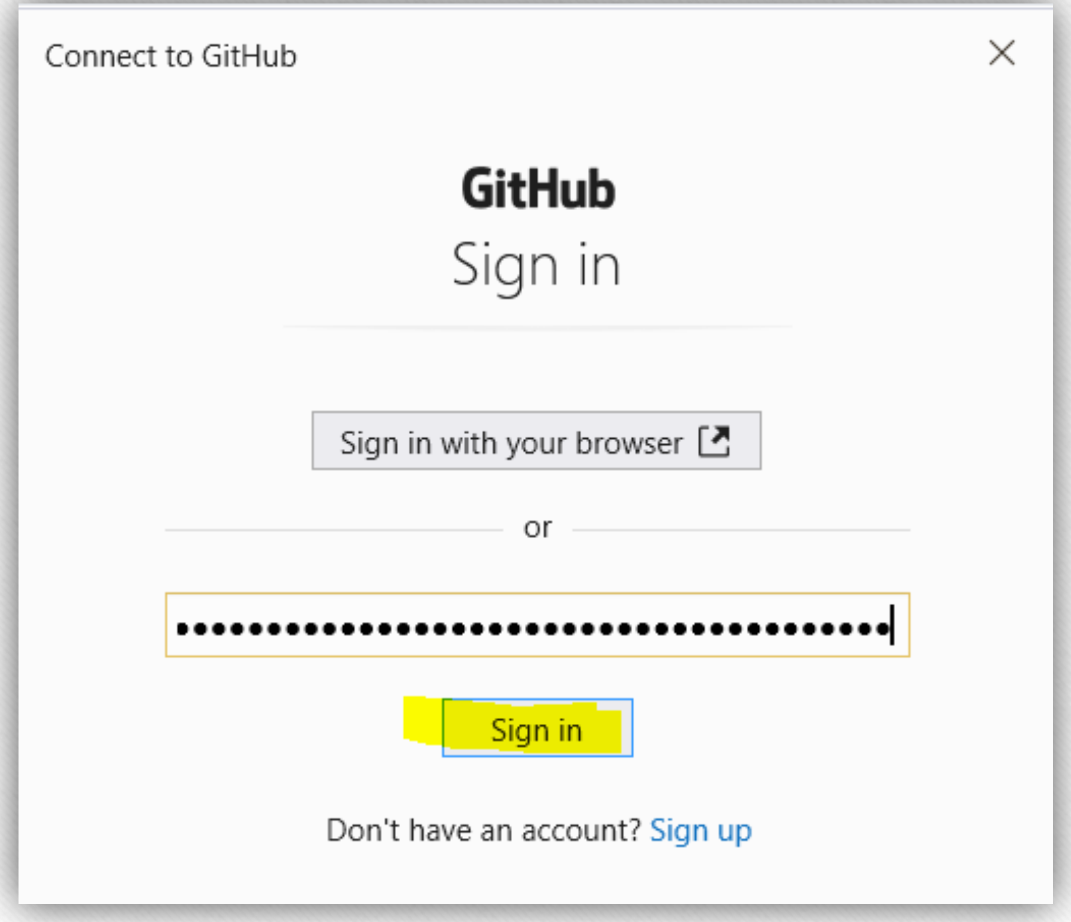


```
 ! [remote rejected] main -> main (refusing to allow a Personal Access Token to create or update workflow `.github/workflows/azure.yml` without `workflow` scope)
```



✓ workflow Update GitHub Action workflows

Enter Newly Generated **Access Token** in this window and click on sign in to continue the pushing Changes to remote repository.

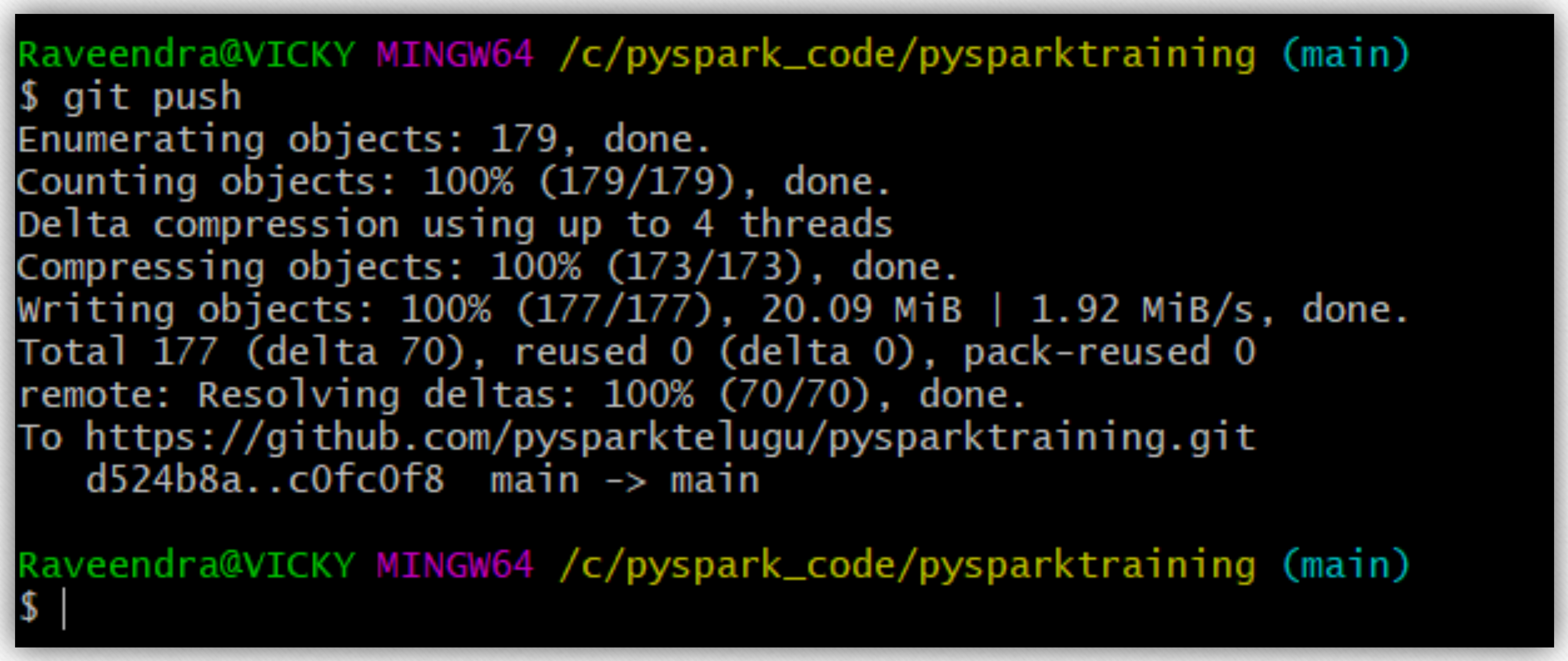

Connect to GitHub

GitHub

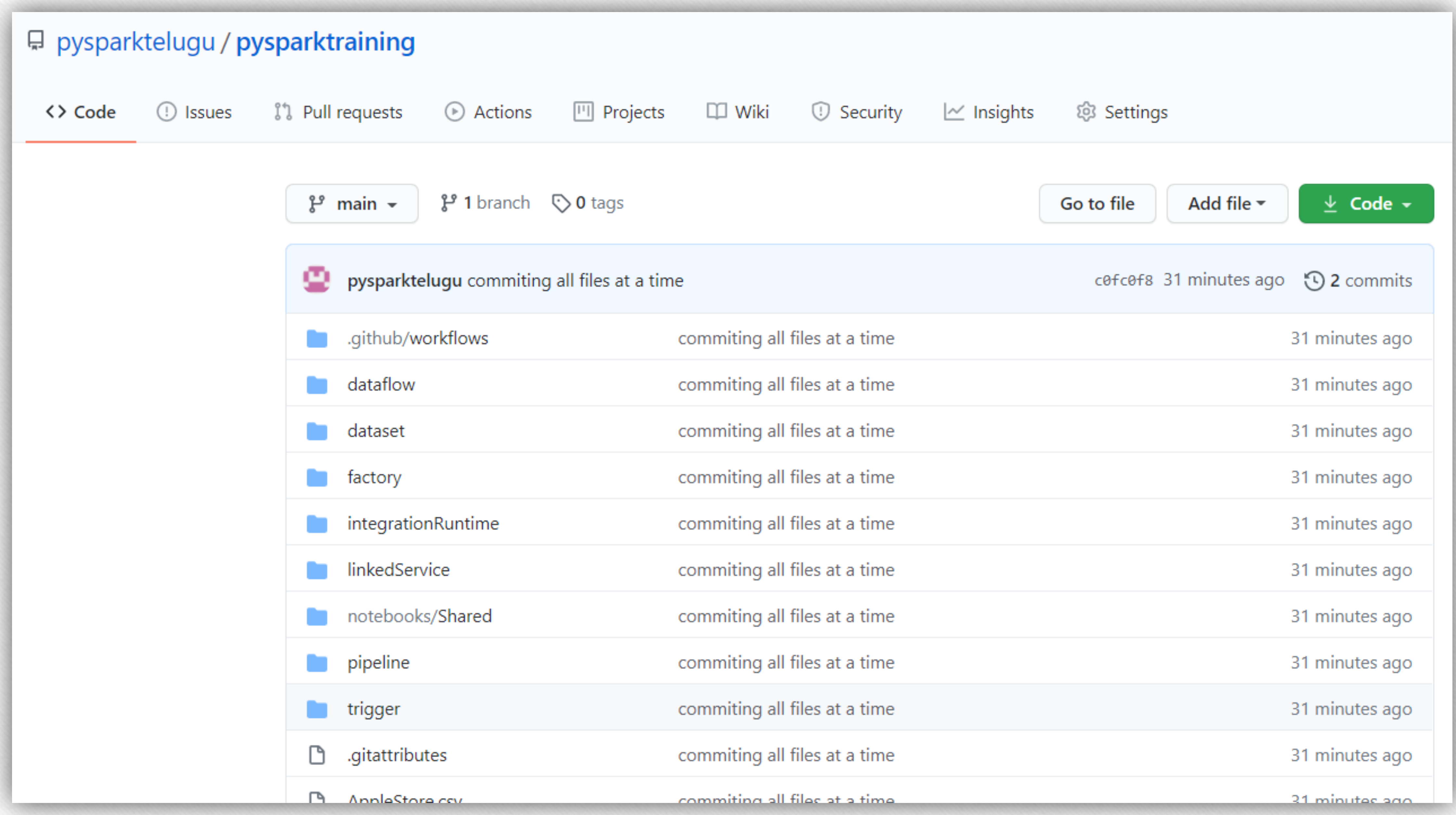
Sign in

Don't have an account? Sign up

Here you can find status after Access Token Authentication it will upload the new changes into
Remote repository.



```
Raveendra@VICKY MINGW64 /c/pyspark_code/pysparktraining (main)
$ git push
Enumerating objects: 179, done.
Counting objects: 100% (179/179), done.
Delta compression using up to 4 threads
Compressing objects: 100% (173/173), done.
Writing objects: 100% (177/177), 20.09 MiB | 1.92 MiB/s, done.
Total 177 (delta 70), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (70/70), done.
To https://github.com/pysparktelugu/pysparktraining.git
   d524b8a..c0fc0f8  main -> main

Raveendra@VICKY MINGW64 /c/pyspark_code/pysparktraining (main)
$ |
```



pysparktelugu / pysparktraining

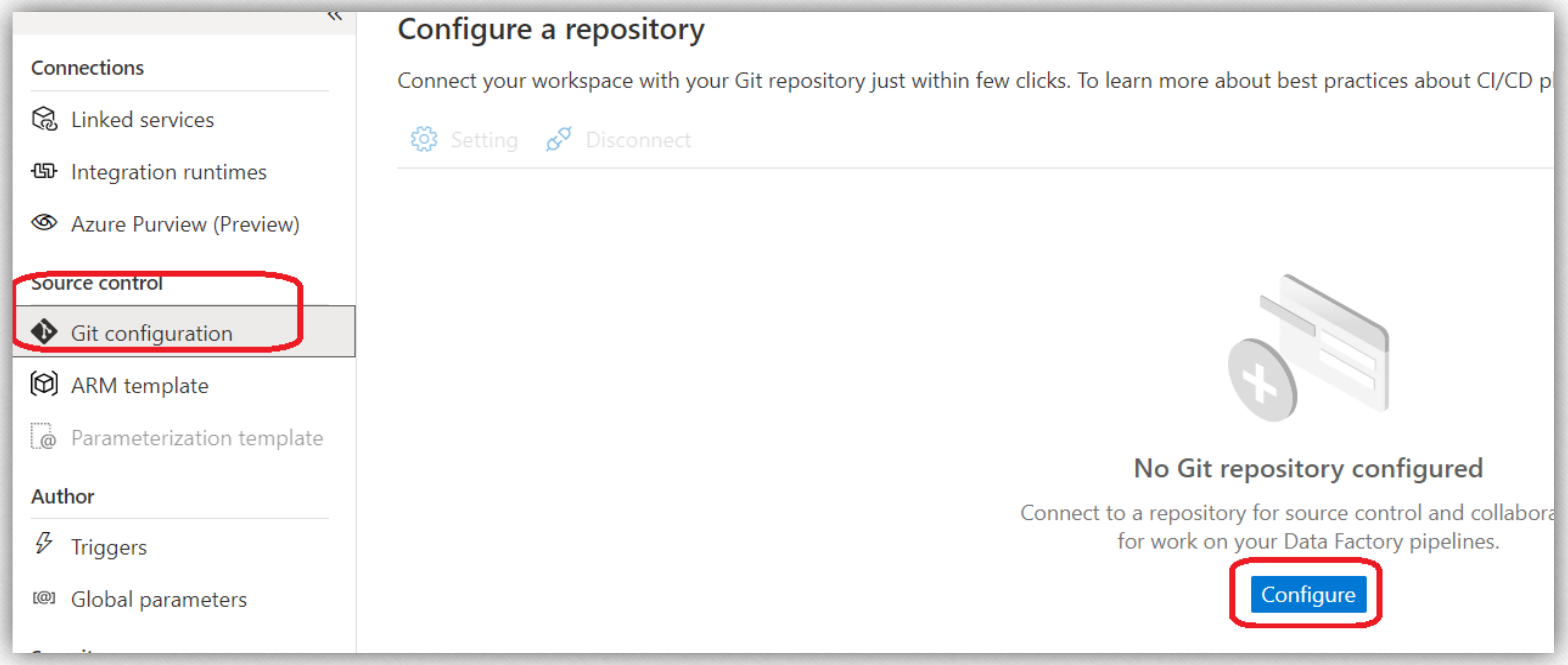
pysparktelugu commiting all files at a time c0fc0f8 31 minutes ago 2 commits

.github/workflows	commiting all files at a time	31 minutes ago
dataflow	commiting all files at a time	31 minutes ago
dataset	commiting all files at a time	31 minutes ago
factory	commiting all files at a time	31 minutes ago
integrationRuntime	commiting all files at a time	31 minutes ago
linkedService	commiting all files at a time	31 minutes ago
notebooks/Shared	commiting all files at a time	31 minutes ago
pipeline	commiting all files at a time	31 minutes ago
trigger	commiting all files at a time	31 minutes ago
.gitattributes	commiting all files at a time	31 minutes ago
AppleStore.csv	commiting all files at a time	31 minutes ago

How To Connect Git in Azure Data Factory?

TECHLAKE
CLOUD IS THE FUTURE.
@trraveendra

Now you can connect newly created Git Repository in Azure Data Factory

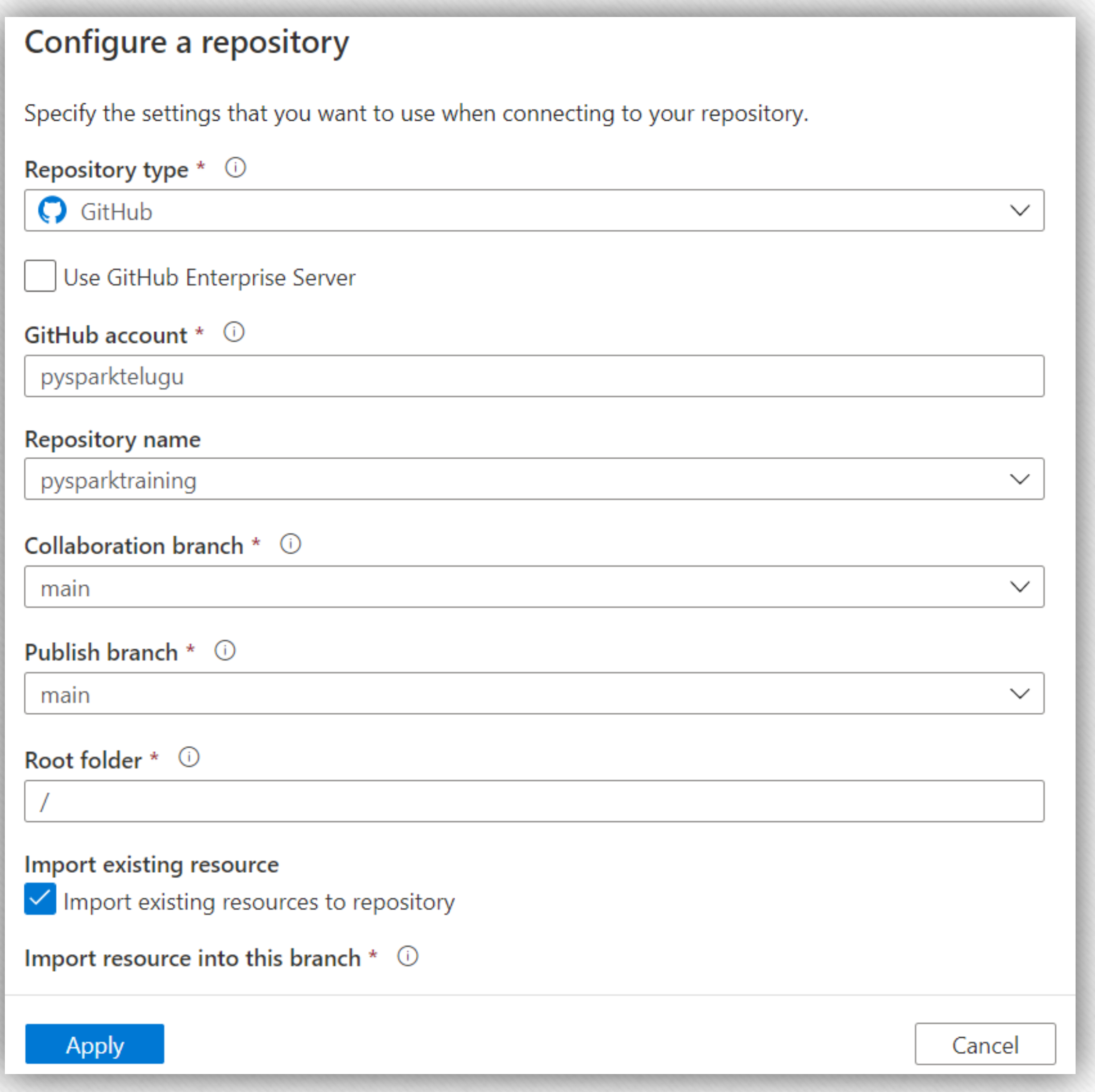


Configure a repository

Connect your workspace with your Git repository just within few clicks. To learn more about best practices about CI/CD pl

No Git repository configured

Connect to a repository for source control and collabora
for work on your Data Factory pipelines.



Configure a repository

Specify the settings that you want to use when connecting to your repository.

Repository type * : GitHub

☐ Use GitHub Enterprise Server

GitHub account * : pysparktelugu

Repository name: pysparktraining

Collaboration branch * : main

Publish branch * : main

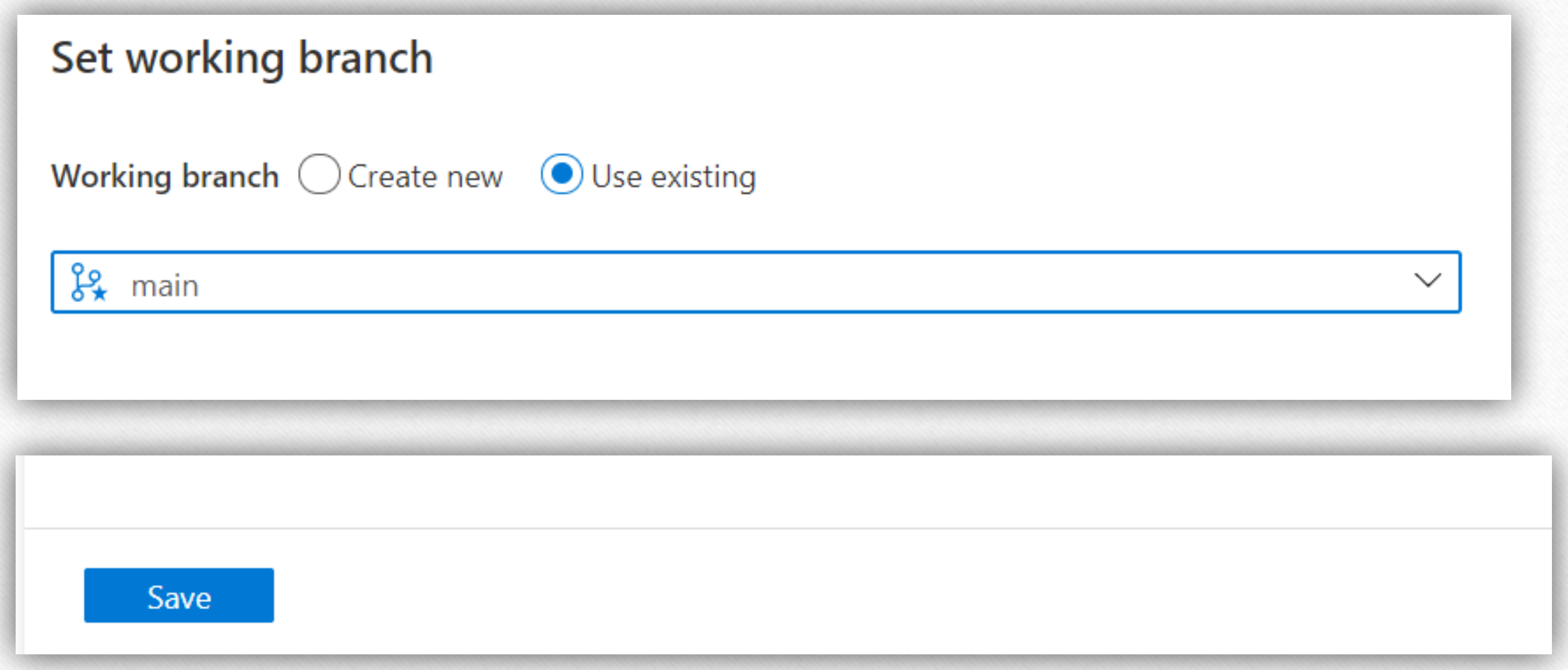
Root folder * : /

Import existing resource

☑ Import existing resources to repository

Import resource into this branch *

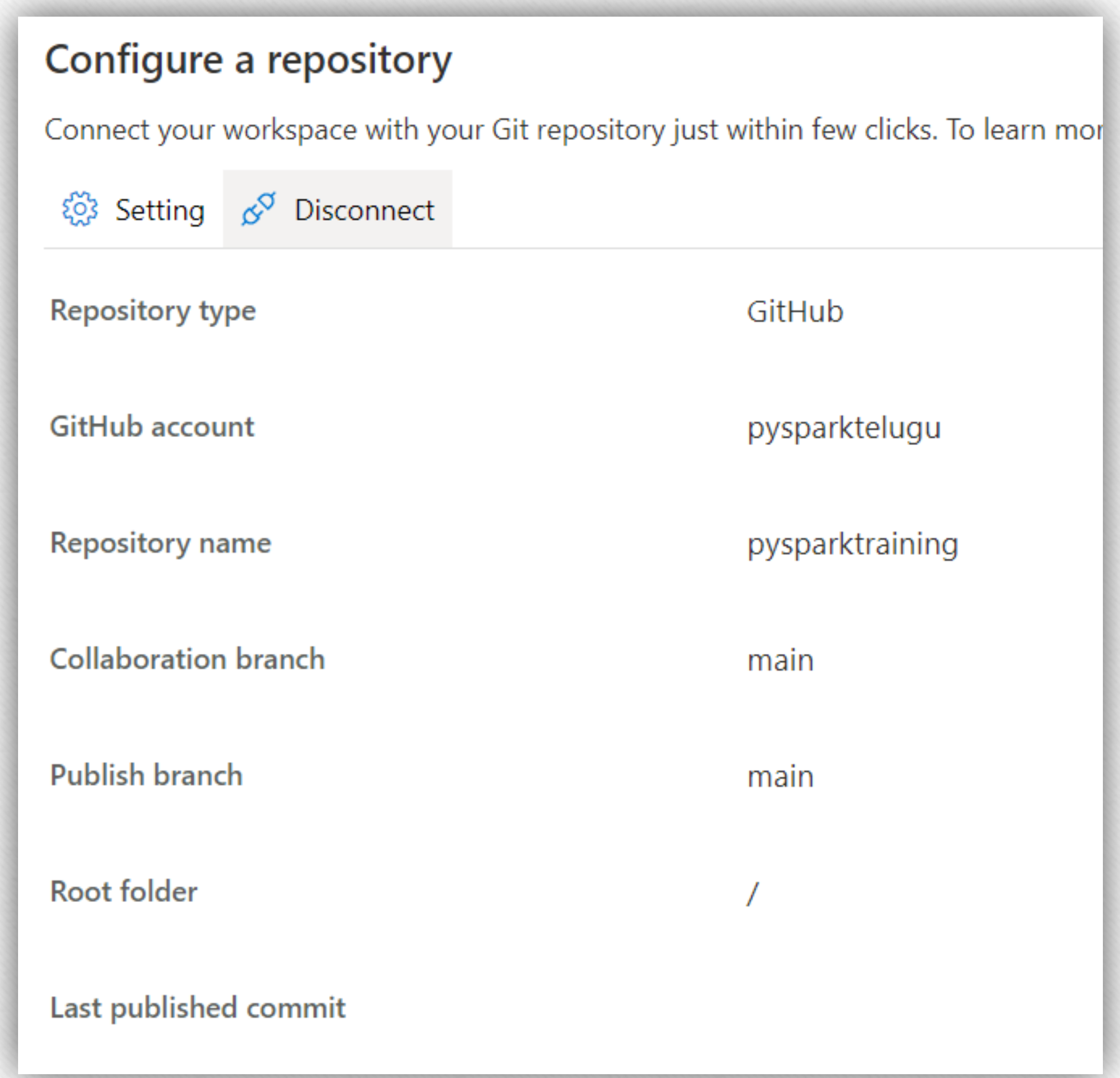
TECHLAKE
CLOUD IS THE FUTURE.
@trraveendra


Set working branch

Working branch ◯ Create new [✓ Use existing]

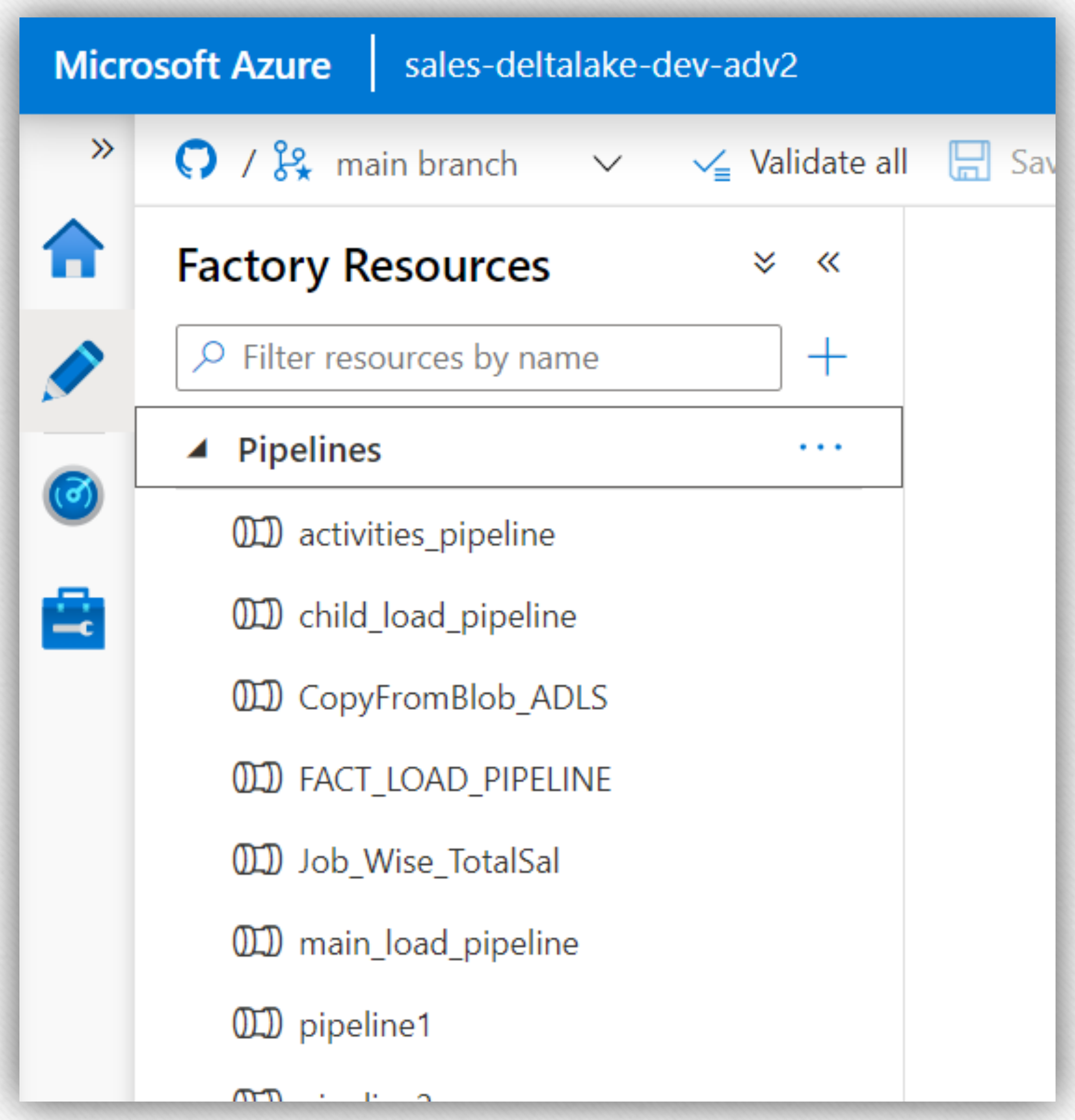
main

TECHLAKE
CLOUD IS THE FUTURE.
@trraveendra


Configure a repository

Connect your workspace with your Git repository just within few clicks. To learn mor

Repository type	GitHub
GitHub account	pysparktelugu
Repository name	pysparktraining
Collaboration branch	main
Publish branch	main
Root folder	/
Last published commit	



activities_pipeline
child_load_pipeline
CopyFromBlob_ADLS
FACT_LOAD_PIPELINE
Job_Wise_TotalSal
main_load_pipeline
pipeline1

All The Best ☺

TECHLAKE

CLOUD IS THE FUTURE.

@trraveendra