PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)

Academic Year: 2024-25

Semester: V

Name: Mayank Salvi

Moodle ID: 22104201

Class / Branch: TEIT

Subject: DevOps Lab

Name of Instructor: Ms. Sujata Oak

Experiment No. 2

Aim: To perform installation of Git and work on local and remote Git repositories.

GIT is a Version Control System (VCS) (aka Revision Control System (RCS), Source Code Manager (SCM)). A VCS serves as a Repository (or repo) of program codes, including all the historical revisions. It records changes to files at so-called commits in a log so that you can recall any file at any commit point.

To issue a command, start a "Terminal" (for Ubuntu/Mac) or "Git Bash" (for Windows):

```
$ git <command> <arguments>
```

The commonly-used commands are:

1. **init**, **clone**, **config**: for starting a Git-managed project.
2. **add**, **mv**, **rm**: for staging file changes.
3. **commit**, **rebase**, **reset**, **tag**:
4. **status**, **log**, **diff**, **grep**, **show**: show status
5. **checkout**, **branch**, **merge**, **push**, **fetch**, **pull**

Getting Started with Local Repo

There are 2 ways to start a Git-managed project:

1. Starting your own project;
2. Cloning an existing project from a GIT host.

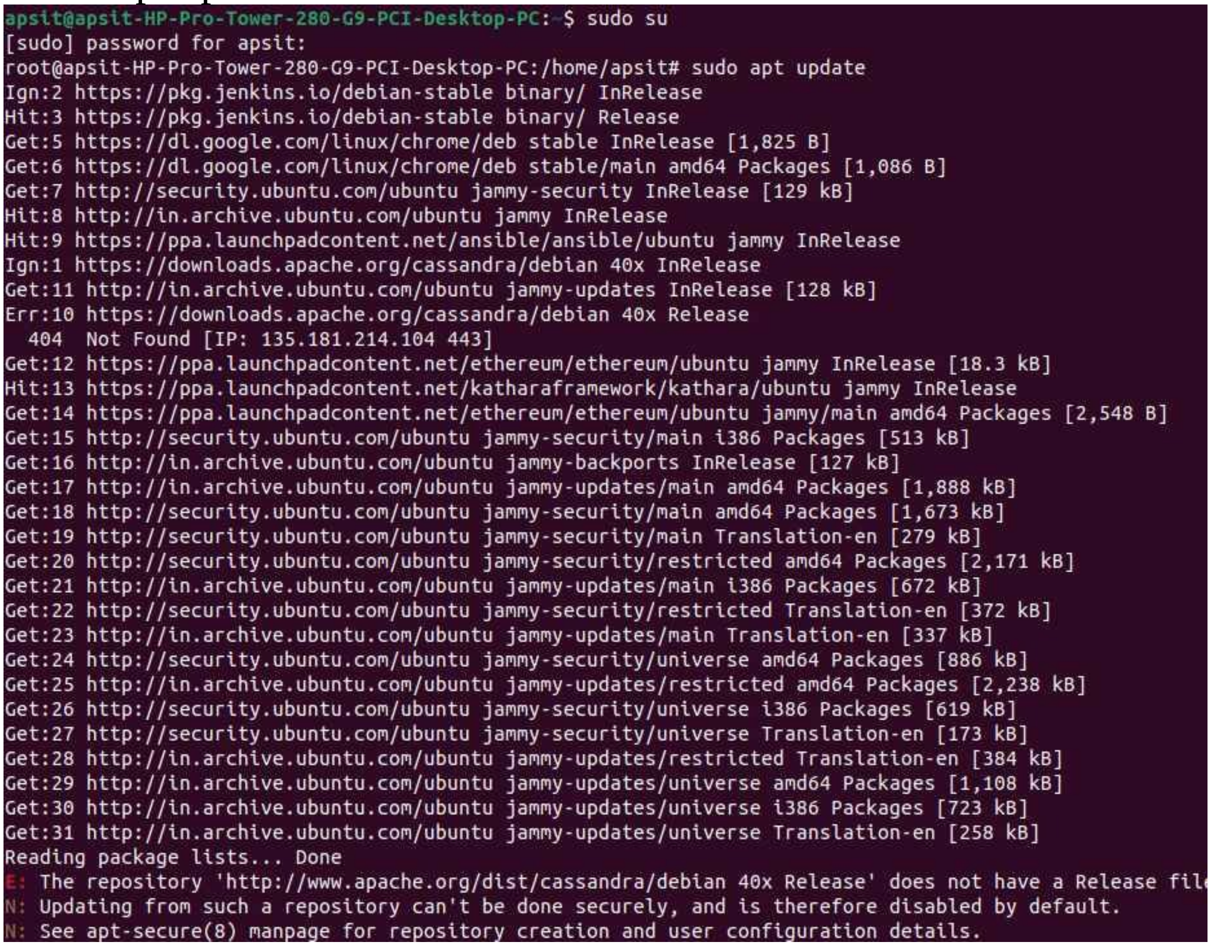
Git uses two stages to commit file changes:

1. "git add <file>" to stage file changes into the staging area, and
2. "git commit" to commit ALL the file changes in the staging area to the local repo.

You need to setup Git on your local machine, as follows:

To update all local package index for ubuntu:

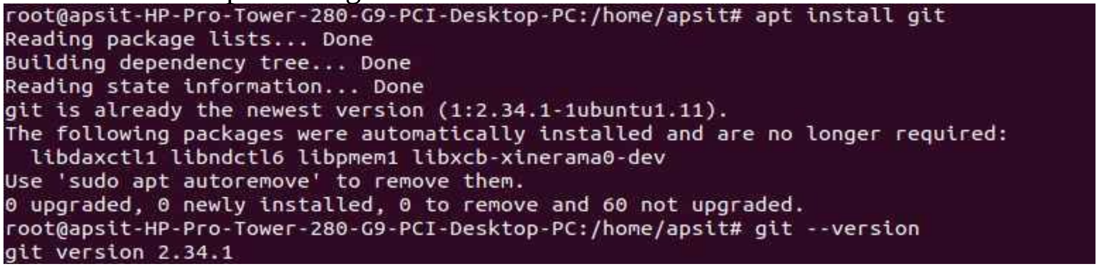
$sudo apt update



```
apsit@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:~$ sudo su
[sudo] password for apsit:
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit# sudo apt update
Ign:2 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:3 https://pkg.jenkins.io/debian-stable binary/ Release
Get:5 https://dl.google.com/linux/chrome/deb stable InRelease [1,825 B]
Get:6 https://dl.google.com/linux/chrome/deb stable/main amd64 Packages [1,086 B]
Get:7 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Hit:8 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Hit:9 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy InRelease
Ign:1 https://downloads.apache.org/cassandra/debian 40x InRelease
Get:11 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Err:10 https://downloads.apache.org/cassandra/debian 40x Release
  404  Not Found [IP: 135.181.214.104 443]
Get:12 https://ppa.launchpadcontent.net/ethereum/ethereum/ubuntu jammy InRelease [18.3 kB]
Hit:13 https://ppa.launchpadcontent.net/katharaframework/kathara/ubuntu jammy InRelease
Get:14 https://ppa.launchpadcontent.net/ethereum/ethereum/ubuntu jammy/main amd64 Packages [2,548 B]
Get:15 http://security.ubuntu.com/ubuntu jammy-security/main i386 Packages [513 kB]
Get:16 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:17 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1,888 kB]
Get:18 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1,673 kB]
Get:19 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [279 kB]
Get:20 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [2,171 kB]
Get:21 http://in.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [672 kB]
Get:22 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [372 kB]
Get:23 http://in.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [337 kB]
Get:24 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [886 kB]
Get:25 http://in.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [2,238 kB]
Get:26 http://security.ubuntu.com/ubuntu jammy-security/universe i386 Packages [619 kB]
Get:27 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [173 kB]
Get:28 http://in.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [384 kB]
Get:29 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1,108 kB]
Get:30 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe i386 Packages [723 kB]
Get:31 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [258 kB]
Reading package lists... Done
E: The repository 'http://www.apache.org/dist/cassandra/debian 40x Release' does not have a Release fil
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
```

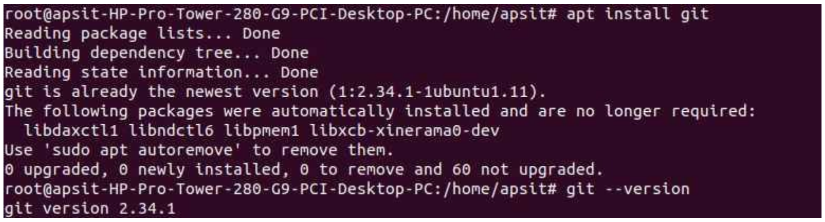
Step1: Download & Install:

- For Ubuntu, issue command "sudo apt-get install git".
- #apt install git



```
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit# apt install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.34.1-1ubuntu1.11).
The following packages were automatically installed and are no longer required:
  libdaxctl1 libndctl6 libpmem1 libxcb-xinerama0-dev
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 60 not upgraded.
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit# git --version
git version 2.34.1
```

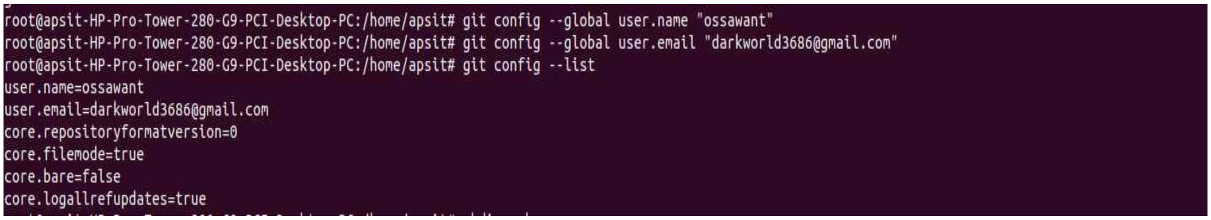
To check version of git:
#git –version



```
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit# apt install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.34.1-1ubuntu1.11).
The following packages were automatically installed and are no longer required:
  libdaxctl1 libndctl6 libpmem1 libxcb-xinerama0-dev
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 60 not upgraded.
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit# git --version
git version 2.34.1
```

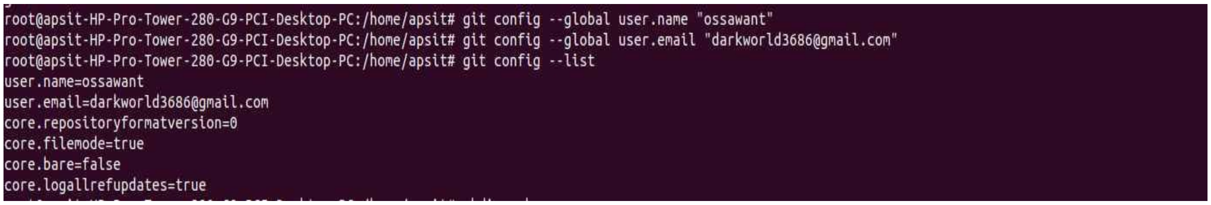
Step 2: Customize and configure your Git Account:

#git config --global user.name “sujataoak799”



```
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit# git config --global user.name "ossawant"
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit# git config --global user.email "darkworld3686@gmail.com"
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit# git config --list
user.name=ossawant
user.email=darkworld3686@gmail.com
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
```

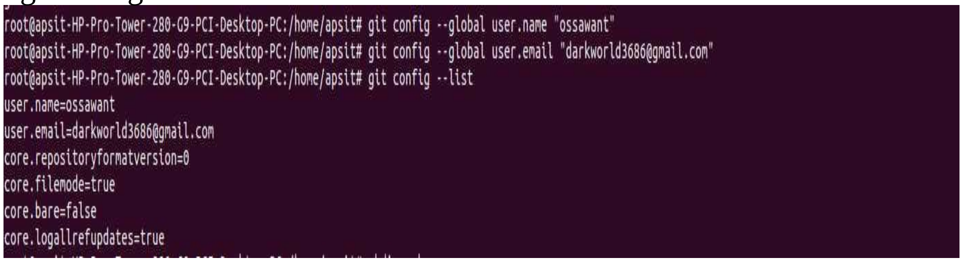
#git config --global user.email sujataoak2021@gmail.com



```
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit# git config --global user.name "ossawant"
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit# git config --global user.email "darkworld3686@gmail.com"
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit# git config --list
user.name=ossawant
user.email=darkworld3686@gmail.com
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
```

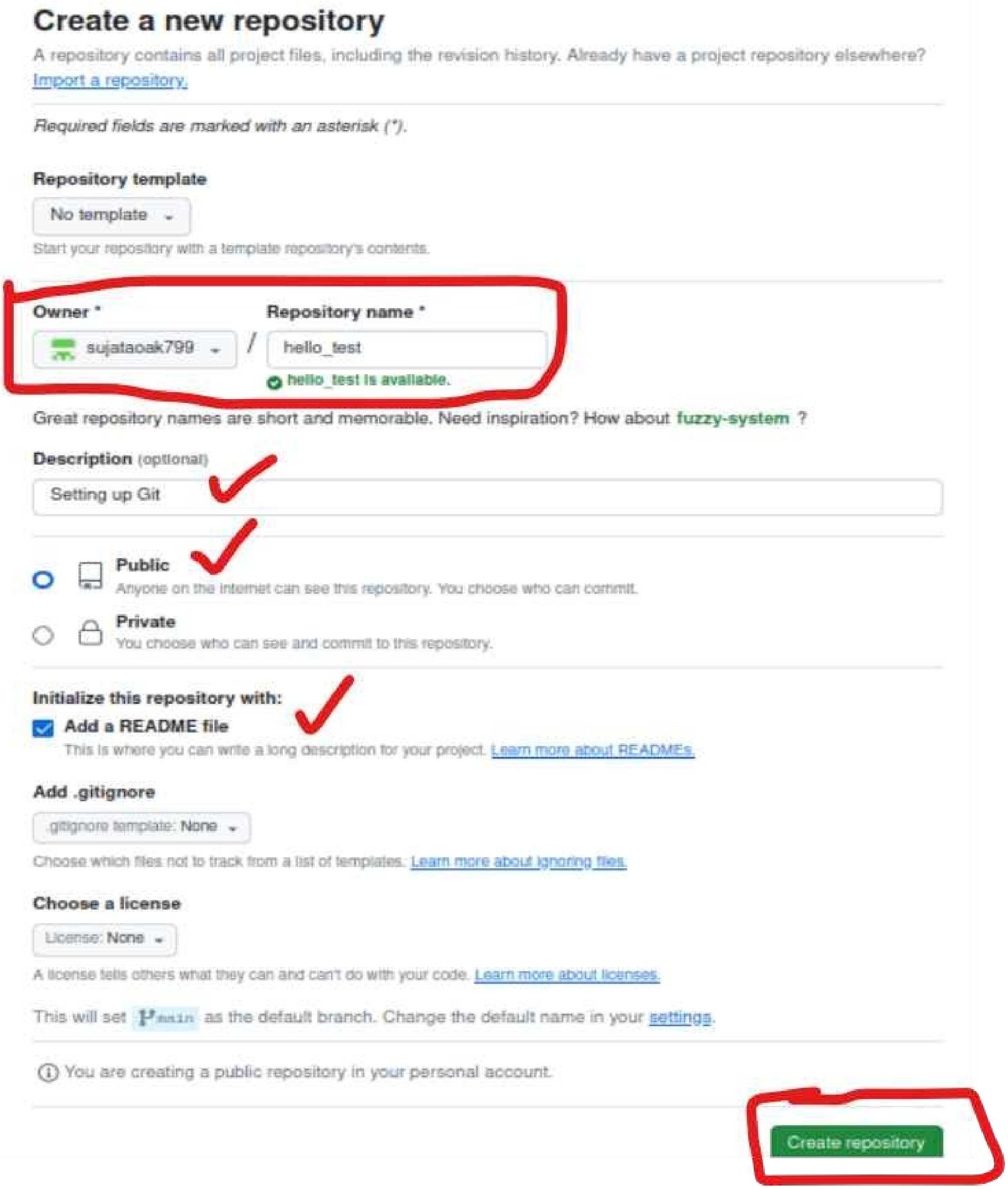
To List Global configuration for Git:

#git config --list



```
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit# git config --global user.name "ossawant"
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit# git config --global user.email "darkworld3686@gmail.com"
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit# git config --list
user.name=ossawant
user.email=darkworld3686@gmail.com
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
```

Step . 3
To Integrate Git account with Github:
Goto www.github.com
Sign-in to your account
Create a Repository



Create a new repository

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

Required fields are marked with an asterisk (*).

Repository template

No template ⌄

Start your repository with a template repository's contents.

Owner * / **Repository name** *

sujataoak799 ⌄ / hello_test

✓ hello_test is available.

Great repository names are short and memorable. Need inspiration? How about **fuzzy-system** ?

Description (optional)

Setting up Git

● **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:

☑ **Add a README file**
This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: None ⌄

Choose which files not to track from a list of templates. Learn more about ignoring files.

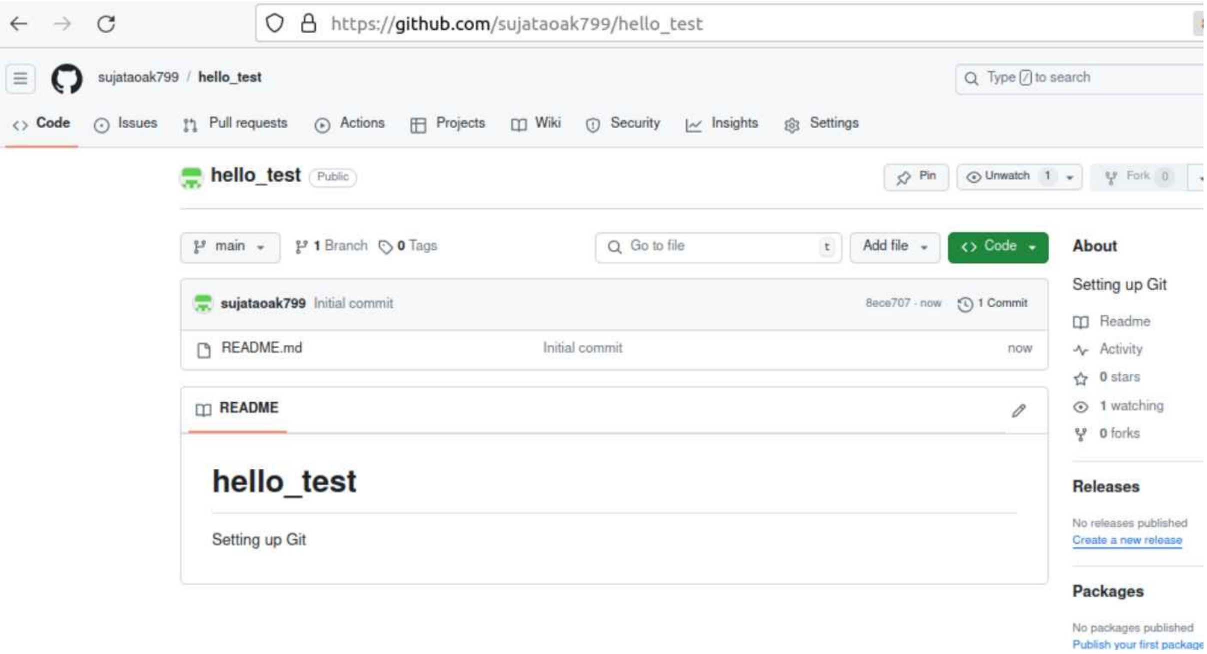
Choose a license

License: None ⌄

A license tells others what they can and can't do with your code. Learn more about licenses.

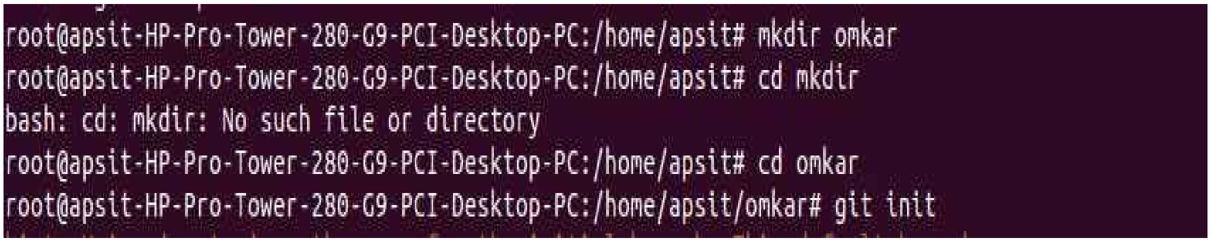
This will set `main` as the default branch. Change the default name in your settings.

ⓘ You are creating a public repository in your personal account.



Step . 3

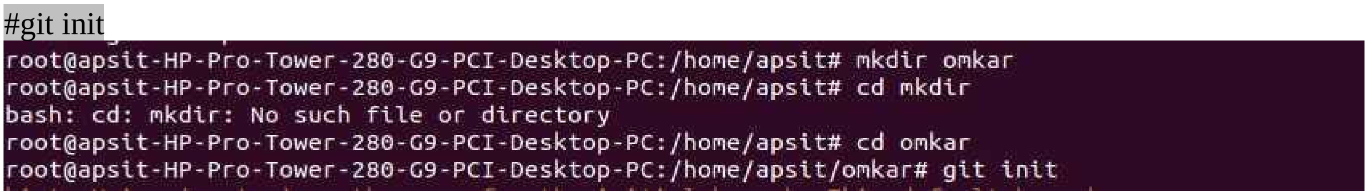
Now Clone the Remote Repository into your Local Repository ie; Ubuntu Operating System:
Firstly create a workspace:



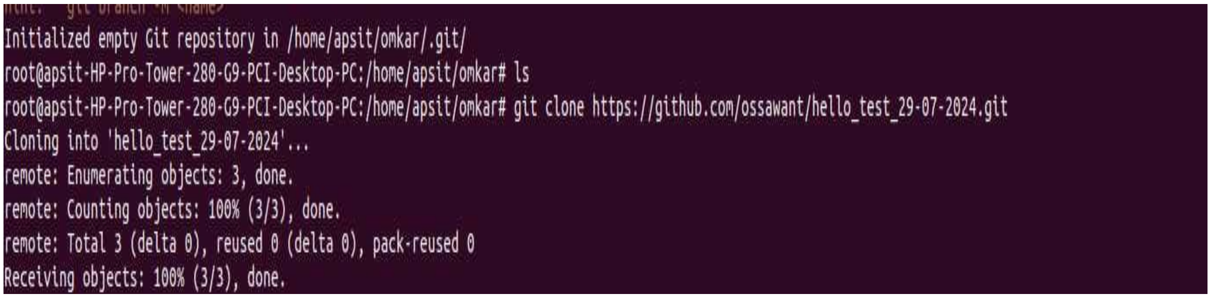
```
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit# mkdir omkar
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit# cd mkdir
bash: cd: mkdir: No such file or directory
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit# cd omkar
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar# git init
```

- Initiate that directory to make it a git repository (.git file must be added inside that folder after initiation)

#git init

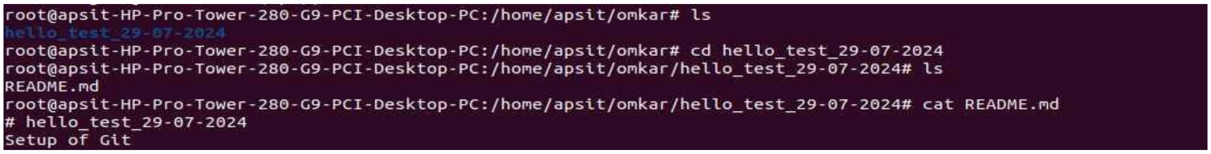


```
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit# mkdir omkar
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit# cd mkdir
bash: cd: mkdir: No such file or directory
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit# cd omkar
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar# git init
```



```
hint:   git branch -m <name>
Initialized empty Git repository in /home/apsit/omkar/.git/
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar# ls
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar# git clone https://github.com/ossawant/hello_test_29-07-2024.git
Cloning into 'hello_test_29-07-2024'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

Now Create a python file hello.py in Local Repository:

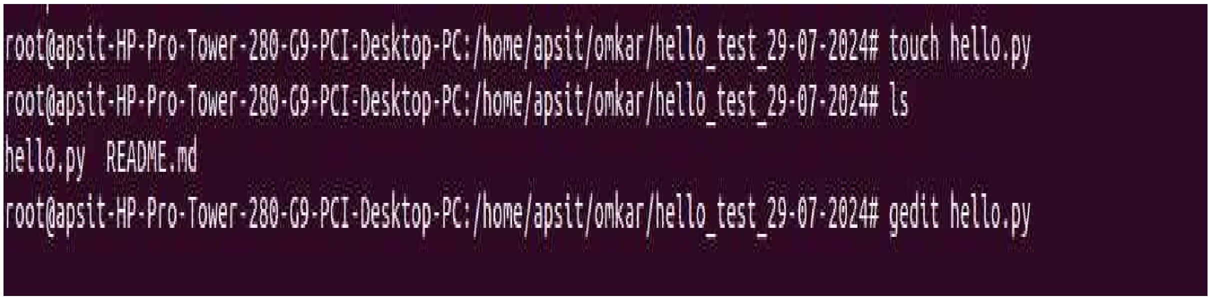


```
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar# ls
hello_test_29-07-2024
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar# cd hello_test_29-07-2024
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar/hello_test_29-07-2024# ls
README.md
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar/hello_test_29-07-2024# cat README.md
# hello_test_29-07-2024
Setup of Git
```

```python
# factorial of given number
def factorial(n):
     if n < 0:
          return 0
     elif n == 0 or n == 1:
          return 1
     else:
          fact = 1
          while(n > 1):
               fact *= n
               n -= 1
          return fact

# Driver Code
num = 5
print("Factorial of",num,"is",
factorial(num))
```

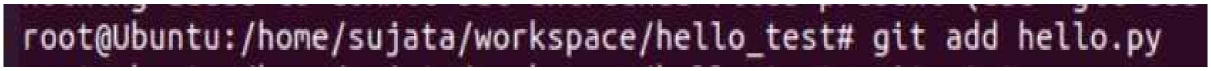
Step 4: Now we will apply some git commands to add, commit and push hello.py file to remote repository: Firstly To View Uncommitted File:

#git status

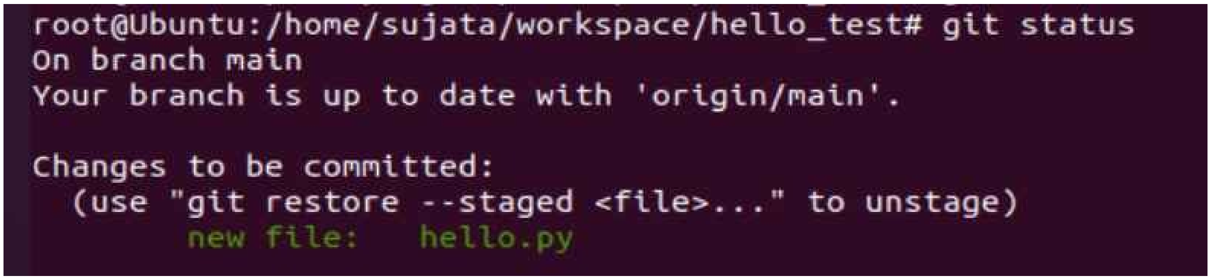


```
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar/hello_test_29-07-2024# touch hello.py
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar/hello_test_29-07-2024# ls
hello.py  README.md
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar/hello_test_29-07-2024# gedit hello.py
```

The file hello.py shown in red color is untracked (it means not tracked by git till now). So to add to the git versioning : git add <filename>



```
root@Ubuntu:/home/sujata/workspace/hello_test# git add hello.py
```

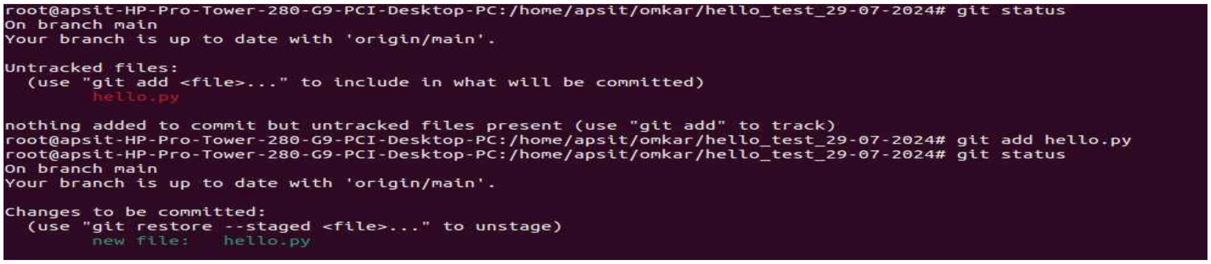


```
root@Ubuntu:/home/sujata/workspace/hello_test# git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   hello.py
```

Once file is added to git, the color changes to green color. But it says changes to be committed

Step 5: To start first commit

git commit -m “First Python File”

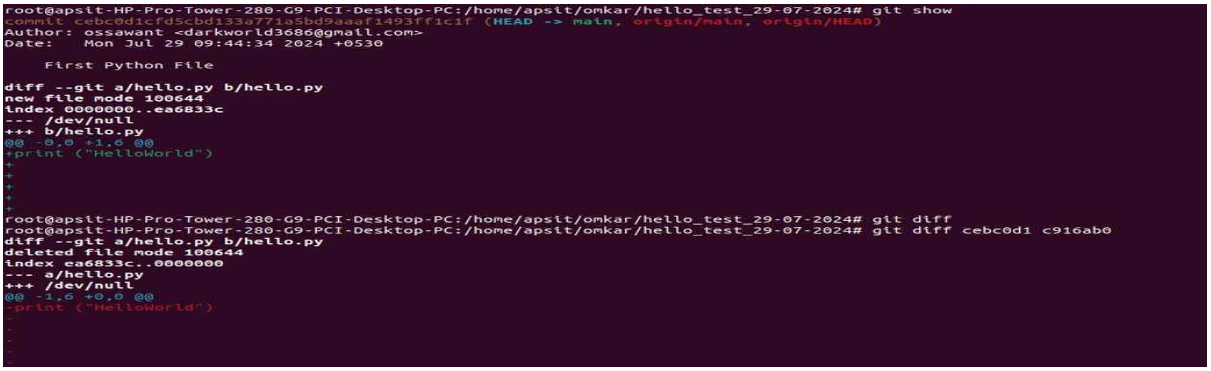


```
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar/hello_test_29-07-2024# git status
On branch main
Your branch is up to date with 'origin/main'.

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

nothing added to commit but untracked files present (use "git add" to track)
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar/hello_test_29-07-2024# git add hello.py
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar/hello_test_29-07-2024# git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   hello.py
```

Step 6: To Push your changes to Github Repository:

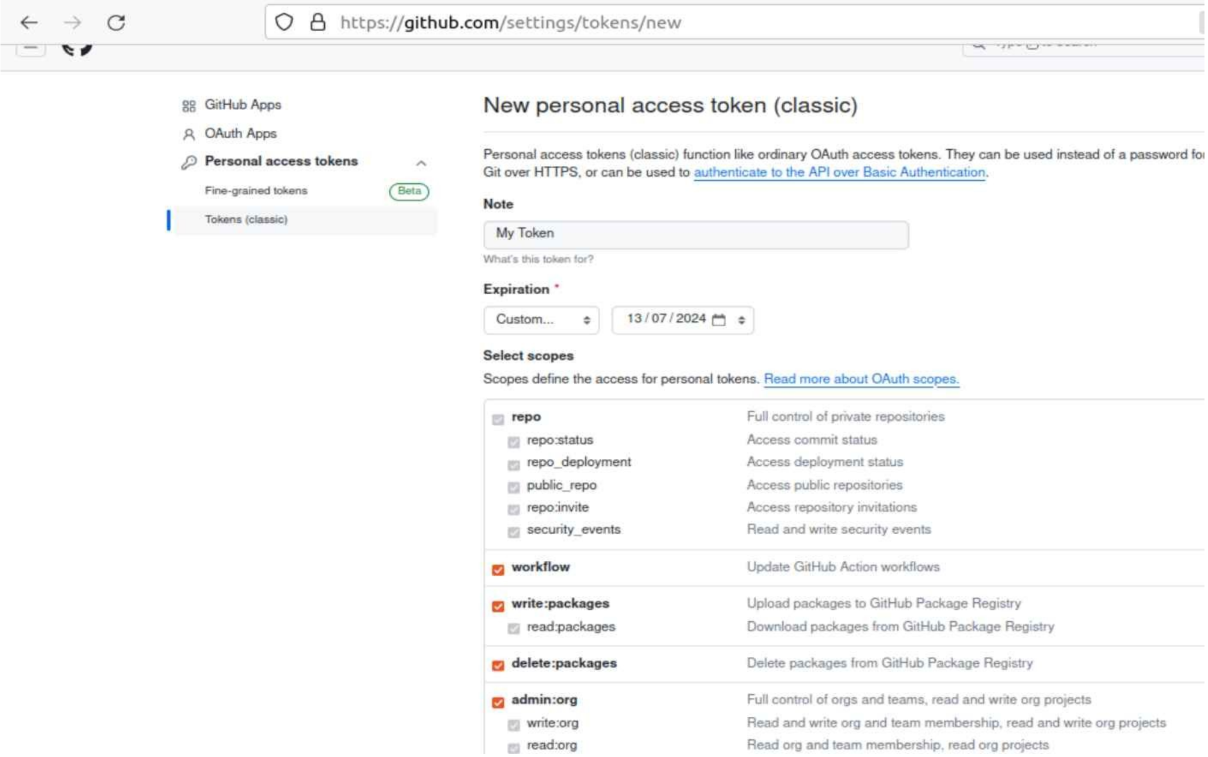
git push origin master or git push origin main

#git push origin main
Enter username and password



```
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar/hello_test_29-07-2024# git show
commit cebc0d1cfd5cbd133a771a5bd9aaaf1493ff1c1f (HEAD -> main, origin/main, origin/HEAD)
Author: ossawant <darkworld3686@gmail.com>
Date:   Mon Jul 29 09:44:34 2024 +0530

    First Python File

diff --git a/hello.py b/hello.py
new file mode 100644
index 0000000..ea6833c
--- /dev/null
+++ b/hello.py
@@ -0,0 +1,6 @@
+print ("HelloWorld")
+
+
+
+
+
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar/hello_test_29-07-2024# git diff
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar/hello_test_29-07-2024# git diff cebc0d1 c916ab0
diff --git a/hello.py b/hello.py
deleted file mode 100644
index ea6833c..0000000
--- a/hello.py
+++ /dev/null
@@ -1,6 +0,0 @@
-print ("HelloWorld")
-
-
-
-
-
```

It says support for password authentication was removed on August 13, 2021. So we need a new kind of method:
Goto your Github Account User Profile Settings Developer Settings Personal Access Token Token(classic) Generate New token Generate New Token(Classic)



https://github.com/settings/tokens/new

New personal access token (classic)

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

Note

My Token

What's this token for?

Expiration *

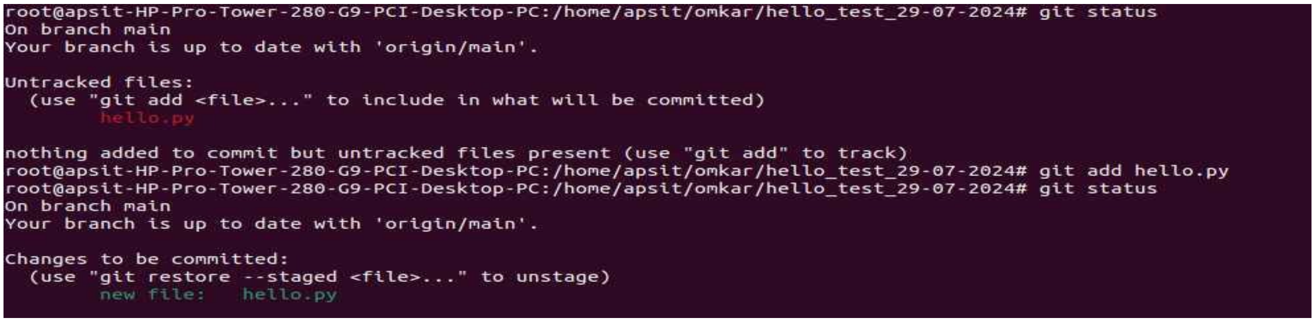
Custom... 13 / 07 / 2024

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
☑ write:packages	Upload packages to GitHub Package Registry
☑ read:packages	Download packages from GitHub Package Registry
☑ delete:packages	Delete packages from GitHub Package Registry
☑ admin:org	Full control of orgs and teams, read and write org projects
☑ write:org	Read and write org and team membership, read and write org projects
☑ read:org	Read org and team membership, read org projects

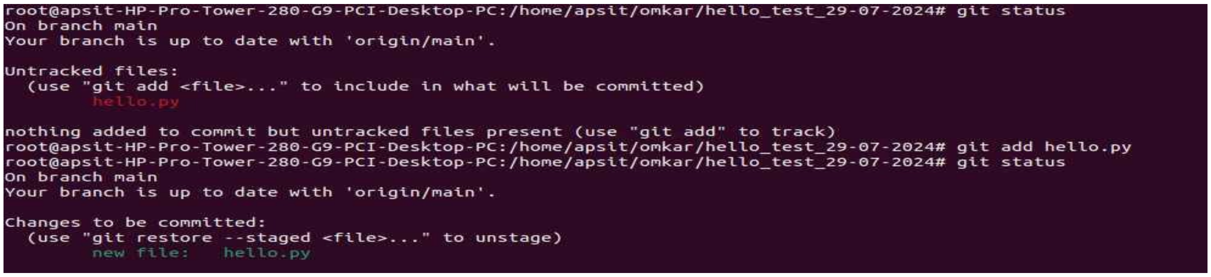
Click on Generate Token
Copy the token and paste it some location:
#git remote set-url origin https://tokenhere@github.com/user_name/repo_name



```
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar/hello_test_29-07-2024# git status
On branch main
Your branch is up to date with 'origin/main'.

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

nothing added to commit but untracked files present (use "git add" to track)
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar/hello_test_29-07-2024# git add hello.py
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar/hello_test_29-07-2024# git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   hello.py
```

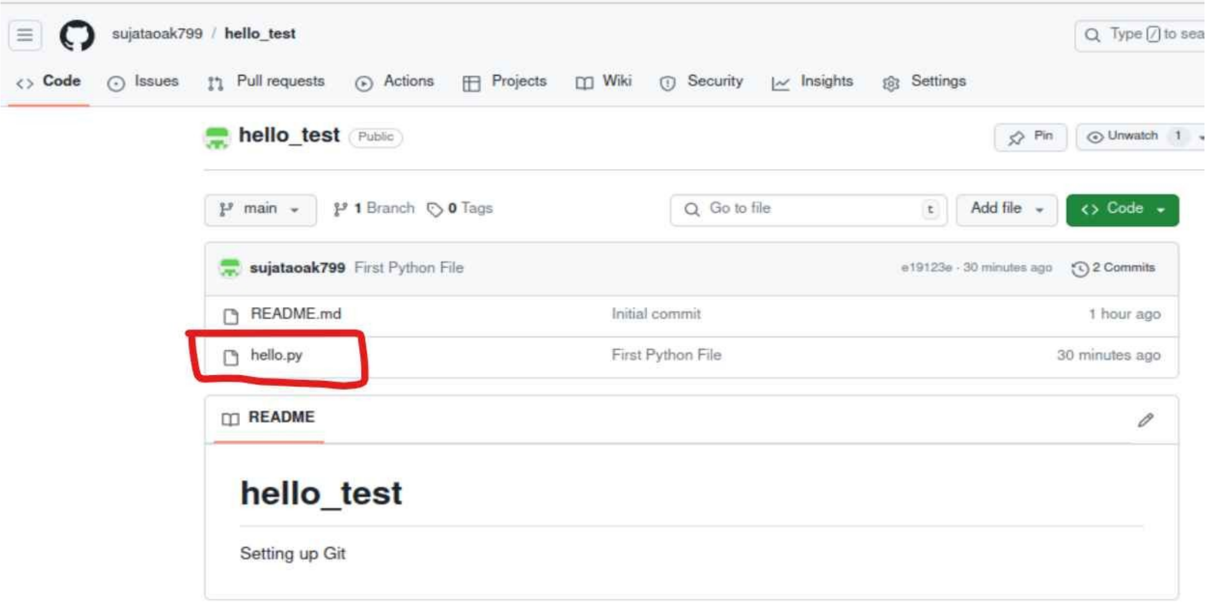
Again try to Push your changes to Github Repository:

#git push origin main



```
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar/hello_test_29-07-2024# git status
On branch main
Your branch is up to date with 'origin/main'.

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

nothing added to commit but untracked files present (use "git add" to track)
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar/hello_test_29-07-2024# git add hello.py
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar/hello_test_29-07-2024# git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   hello.py
```

This time it is successfully implemented.
Goto Remote Repository and see the file hello.py



hello_test Public

sujataoak799 First Python File e19123e · 30 minutes ago 2 Commits

README.md	Initial commit	1 hour ago
hello.py	First Python File	30 minutes ago

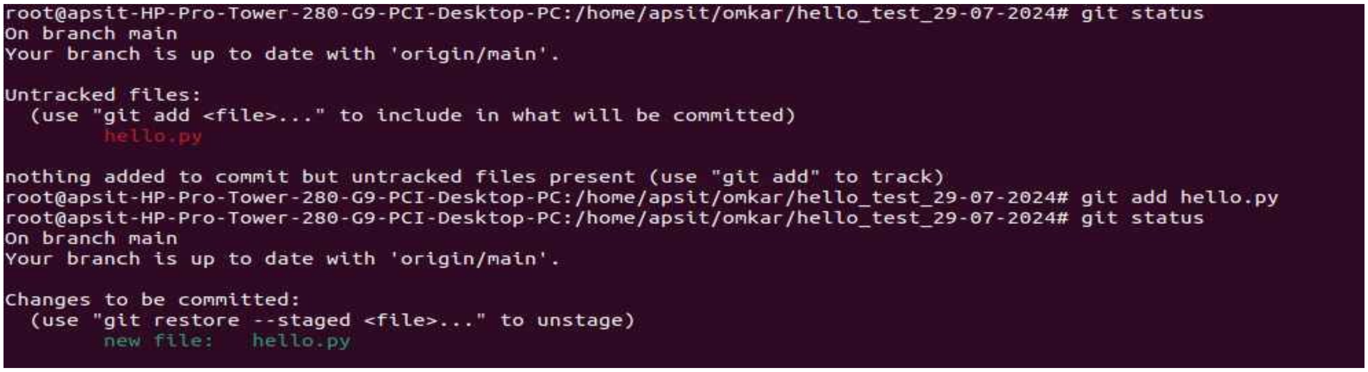
README

hello_test

Setting up Git

Step7 : To see the logs in oneline like username, email -id, date, time of creation.

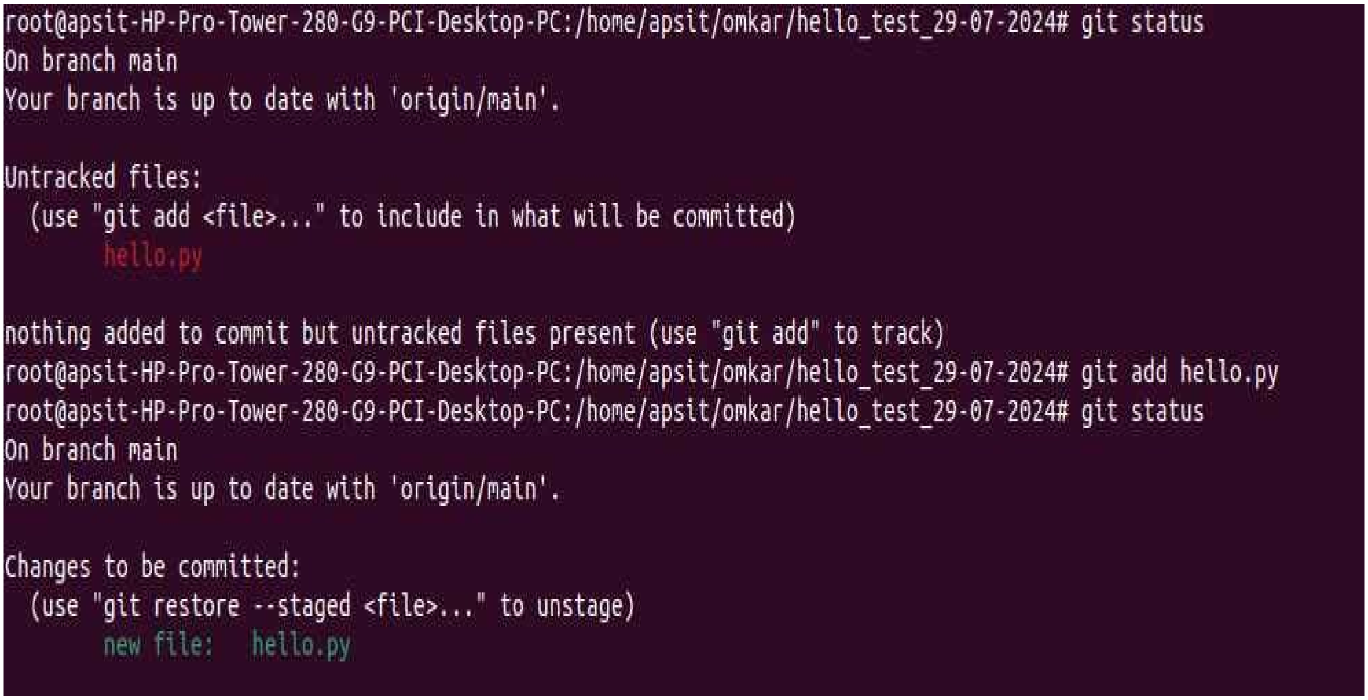
git log
 or
git log –oneline



```
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar/hello_test_29-07-2024# git status
On branch main
Your branch is up to date with 'origin/main'.

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

nothing added to commit but untracked files present (use "git add" to track)
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar/hello_test_29-07-2024# git add hello.py
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar/hello_test_29-07-2024# git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   hello.py
```

Step 8: To show repository id and other detail

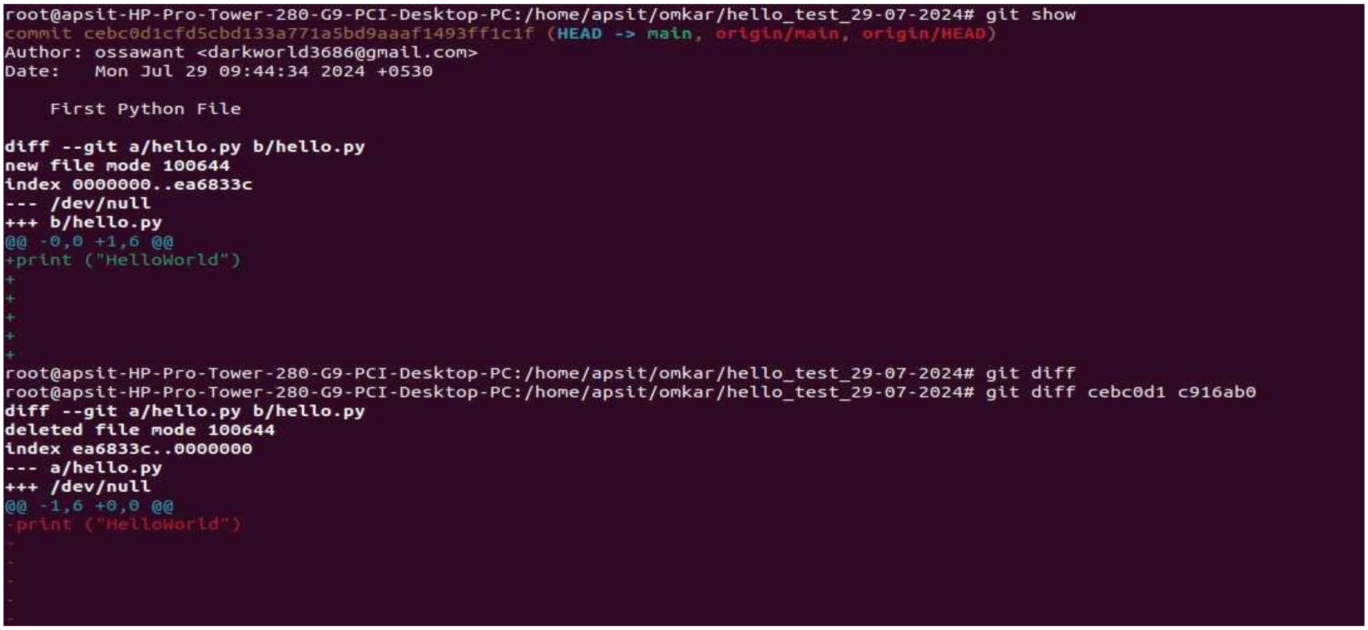
#git show



```
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar/hello_test_29-07-2024# git status
On branch main
Your branch is up to date with 'origin/main'.

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

nothing added to commit but untracked files present (use "git add" to track)
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar/hello_test_29-07-2024# git add hello.py
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar/hello_test_29-07-2024# git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   hello.py
```

Step 09: To see the difference in the content of file between first and second commit.

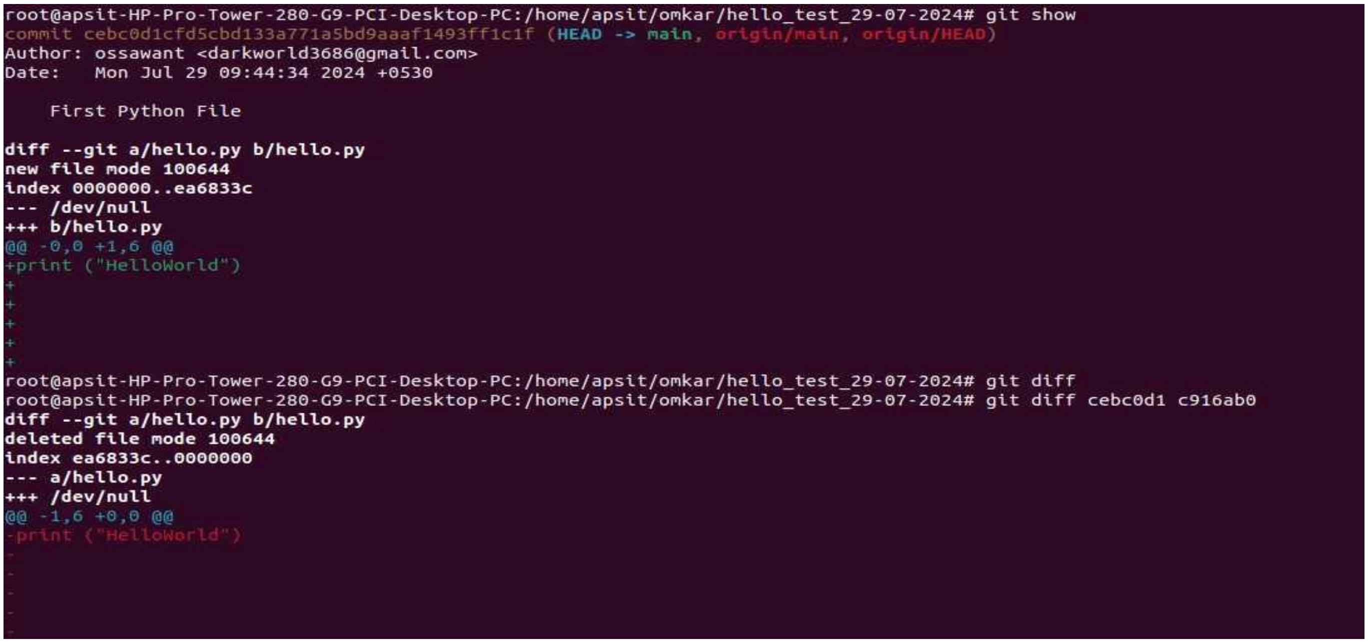
#git diff



```
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar/hello_test_29-07-2024# git show
commit cebc0d1cfd5cbd133a771a5bd9aaaf1493ff1c1f (HEAD -> main, origin/main, origin/HEAD)
Author: ossawant <darkworld3686@gmail.com>
Date:   Mon Jul 29 09:44:34 2024 +0530

    First Python File

diff --git a/hello.py b/hello.py
new file mode 100644
index 0000000..ea6833c
--- /dev/null
+++ b/hello.py
@@ -0,0 +1,6 @@
+print ("HelloWorld")
+
+
+
+
+
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar/hello_test_29-07-2024# git diff
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar/hello_test_29-07-2024# git diff cebc0d1 c916ab0
diff --git a/hello.py b/hello.py
deleted file mode 100644
index ea6833c..0000000
--- a/hello.py
+++ /dev/null
@@ -1,6 +0,0 @@
-print ("HelloWorld")
-
-
-
-
```

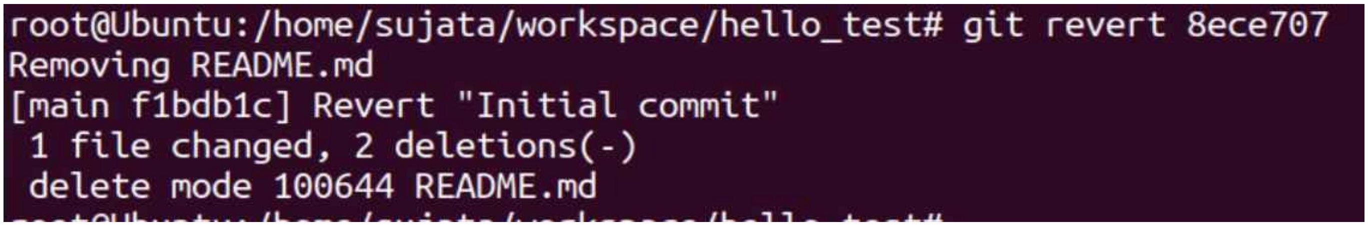
Step 10: Creating a latter commit and reverting back to see the initial/original content

git revert<secondcommitID>



```
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar/hello_test_29-07-2024# git show
commit cebc0d1cfd5cbd133a771a5bd9aaaf1493ff1c1f (HEAD -> main, origin/main, origin/HEAD)
Author: ossawant <darkworld3686@gmail.com>
Date:   Mon Jul 29 09:44:34 2024 +0530

    First Python File

diff --git a/hello.py b/hello.py
new file mode 100644
index 0000000..ea6833c
--- /dev/null
+++ b/hello.py
@@ -0,0 +1,6 @@
+print ("HelloWorld")
+
+
+
+
+
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar/hello_test_29-07-2024# git diff
root@apsit-HP-Pro-Tower-280-G9-PCI-Desktop-PC:/home/apsit/omkar/hello_test_29-07-2024# git diff cebc0d1 c916ab0
diff --git a/hello.py b/hello.py
deleted file mode 100644
index ea6833c..0000000
--- a/hello.py
+++ /dev/null
@@ -1,6 +0,0 @@
-print ("HelloWorld")
-
-
-
-
```

Save it



```
root@Ubuntu:/home/sujata/workspace/hello_test# git revert 8ece707
Removing README.md
[main f1bdb1c] Revert "Initial commit"
 1 file changed, 2 deletions(-)
 delete mode 100644 README.md
root@Ubuntu:/home/sujata/workspace/hello_test#
```

Conclusion:

In this experiment, we understood the use case of Version Control System, its benefits in real time scenario which provides a application of reverting the changes when people are in working in a collaborating environment. Different commands were used for the same such as revert(by using its id),diff for displaying the changes between the initial and latter texts.