What is Git?

-----------------------

Git is a Version controlling tool / SCM tool. (Source code mgmt tool). Version control systems are tools that help a software team manage changes to source code over time. For almost all software projects, the source code is like the crown jewels - a precious asset whose value must be protected. VCS are sometimes known as SCM (Source Code Management) tool.

Most widely used modern version control system in the world today is Git. Git is a mature, actively maintained open source tool originally developed in 2005 by Linus Torvalds, the famous creator of the Linux operating system kernel.

Two types Version Controlloing

1) Centrailized Version controlling

2) Distributed Version controlling

Git is Distributed Version controlling.

Version Control System (VCS) is a software that helps software developers to work together and maintain a complete history of their work.

Git is a distributed version control system. It does not depend on central server.  Operations can be performed offline and the changes can be published onto repository.

**Benefits:**

* Free and open source
* Fast and small
* Implicit backup
* Security
* No need of powerful hardware
* Easier branching
* Github.com is a website that hosts git repositories on a remote server

* Hosting repositories on Github facilitates the sharing of codebases among teams by providing a GUI to easily fork or clone repos to a local machine

* By pushing your repositories to Github, you will pretty much automatically create your own developer portfolio as well!

Shape

Description automatically generated with medium confidence

Three main states of a Git:

* **Working Directory –** local disk

where the changes are done

* **Staging Area –** a file containing

information about what will go into

next commit

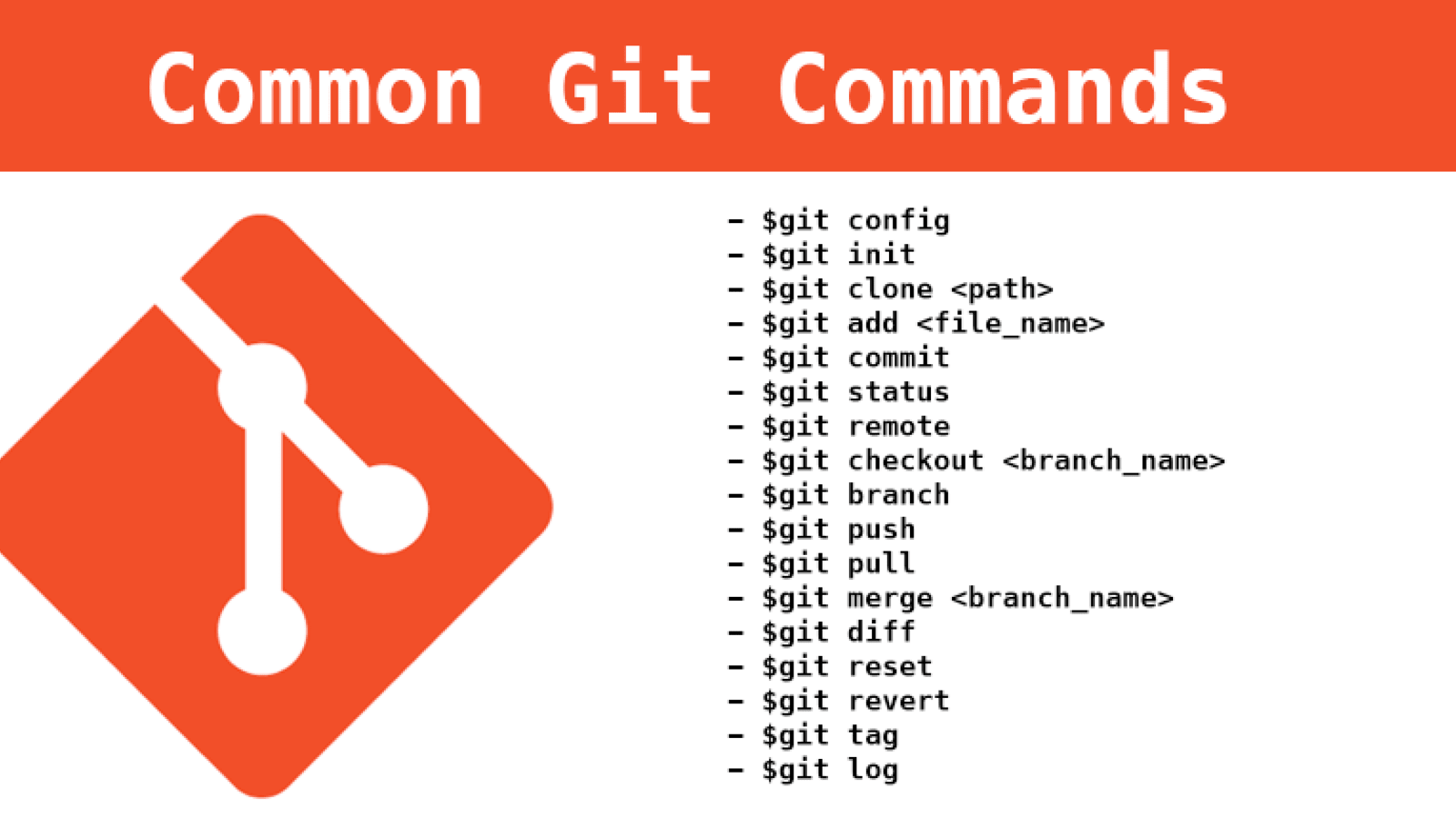
* **Git Directory –**stores metadata and

Object database for the project

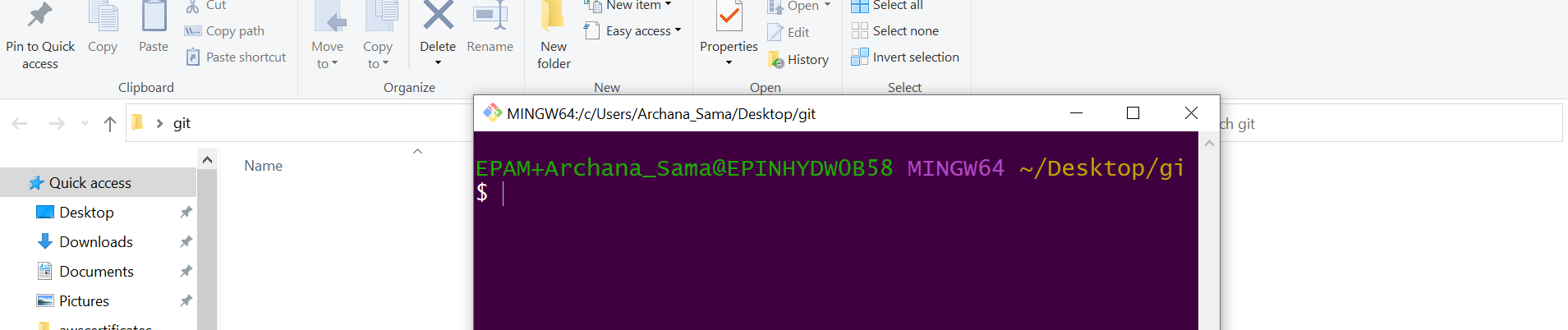
while cloning the repository

Diagram

Description automatically generated



Before git init command



$ git init

Graphical user interface, text, application

Description automatically generated

How to configure username and email for git

$ git config --global user.name "archanareddyse"

$ git config --global user.email “archanareddycse@gmail.com”

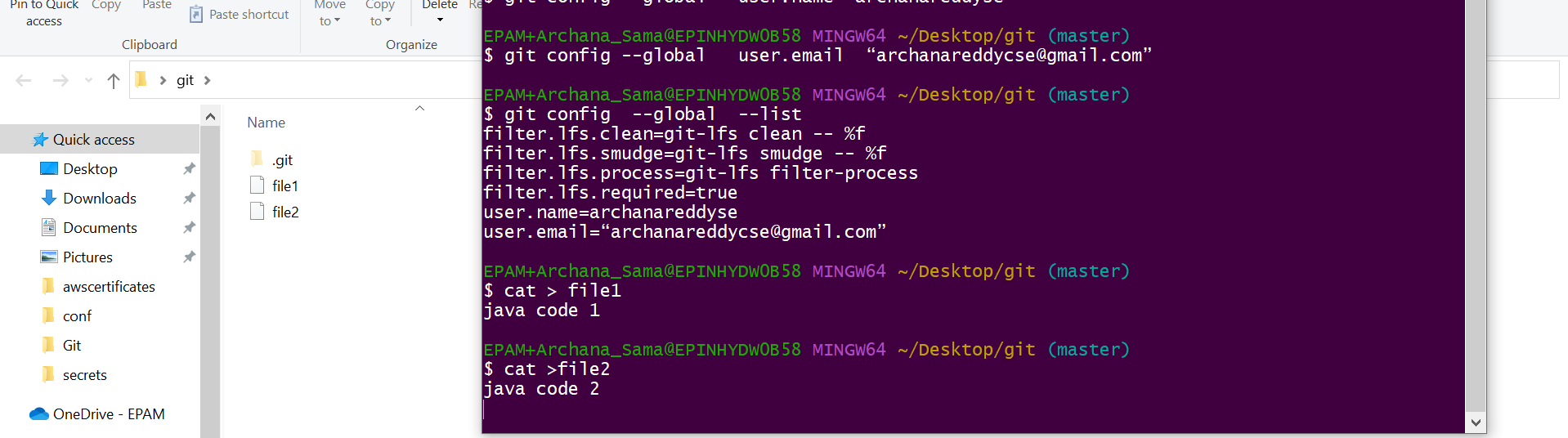
$ git config –global --list

Text

Description automatically generated

Crete the file by file1,file 2

$ cat file1 and $ file2



**Types of files**

Untracked files –can be seen by git status command which shows in red color

Staged files --- files can be can be stagged by git add command

Committed files --- by this command files are stored in local repository

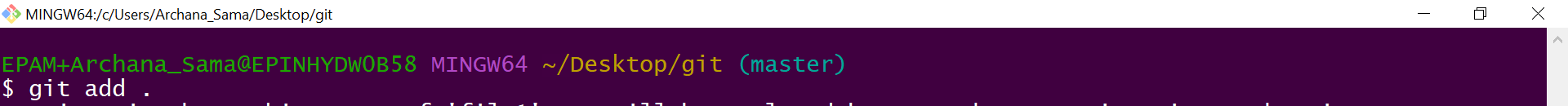
**$ git status**

Text

Description automatically generated

$ git add filename [for single file]

$ git add .[for more than files]



After stagging git status will show us green color

Text

Description automatically generated

$ git commit -m “message for commit”

Text

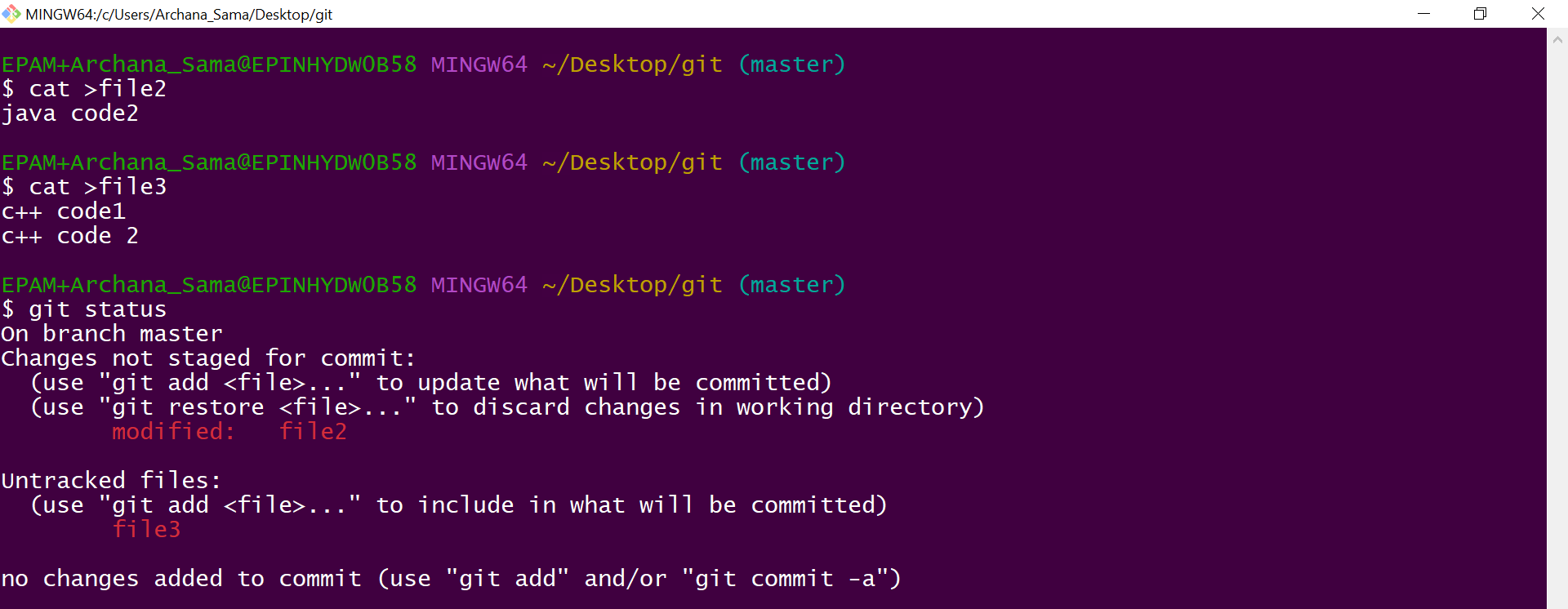
Description automatically generated

$ git log [shows us all committed messages]

Text

Description automatically generated

Now create more files and check the status



Now add the files to stagging area by $ git add.

Text

Description automatically generated

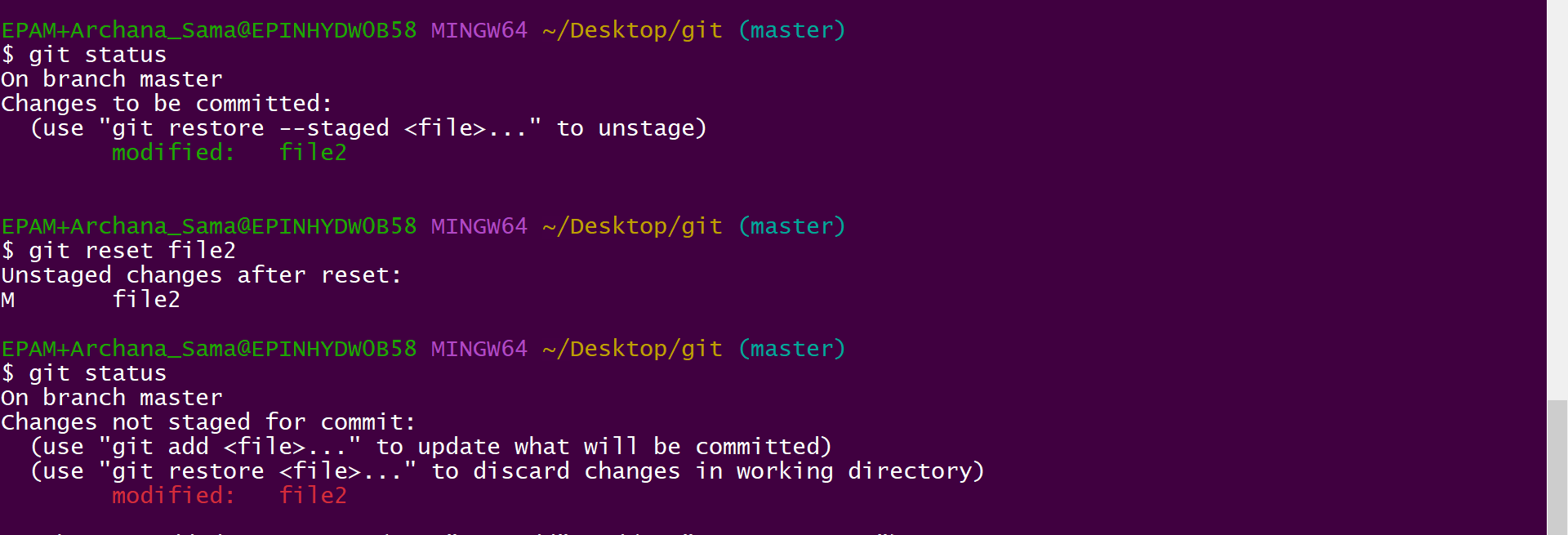
If you want to restore the files from the stagging area which are not committed the command is

$ git restore –staged file2

$ git reset file2

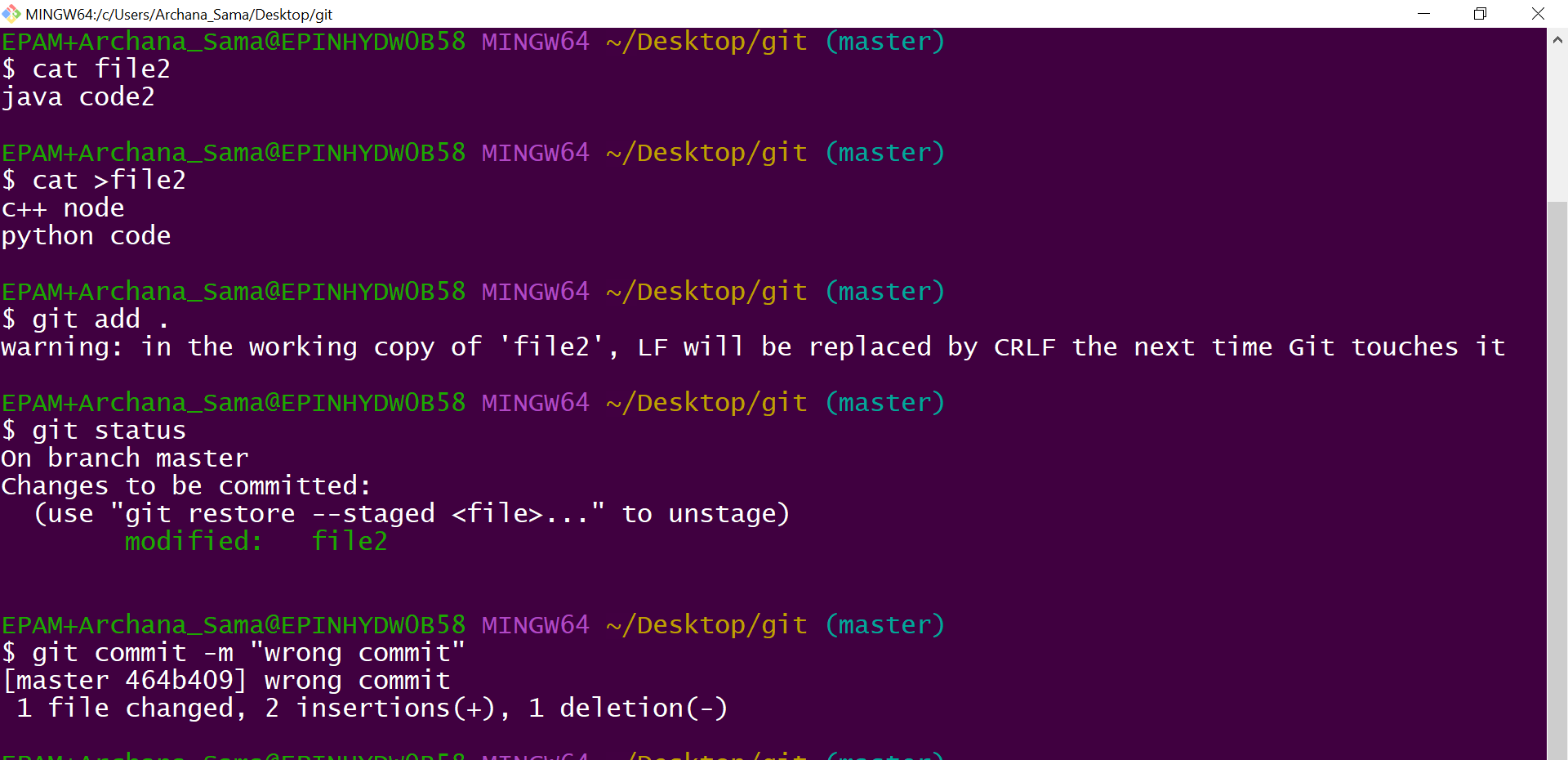


$ git reset filename



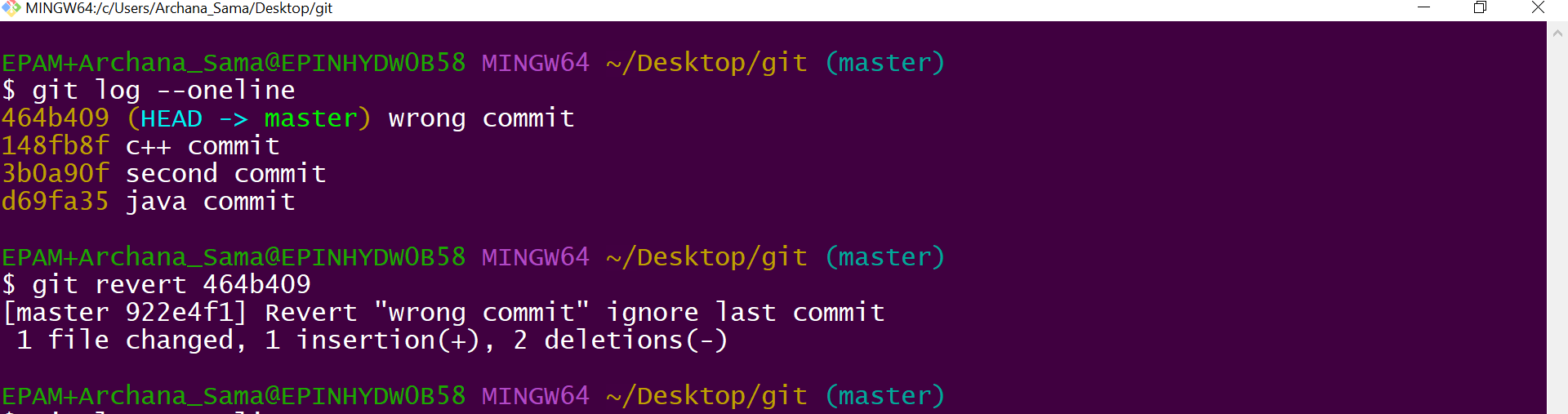
If you want to restore the files which are committed

$ git revert [commit id]



$ git log –oneline [shows all the commit in one single line]

$ git revert [commit id]



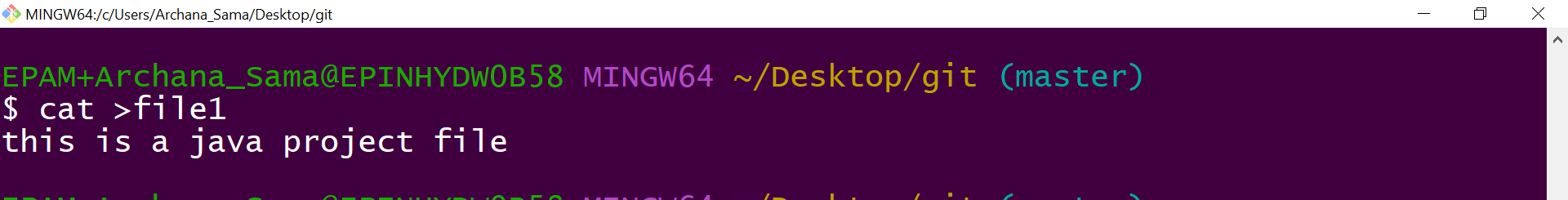
After revert command a file will displayed type message you to type to ignore the commit

:wq

Text

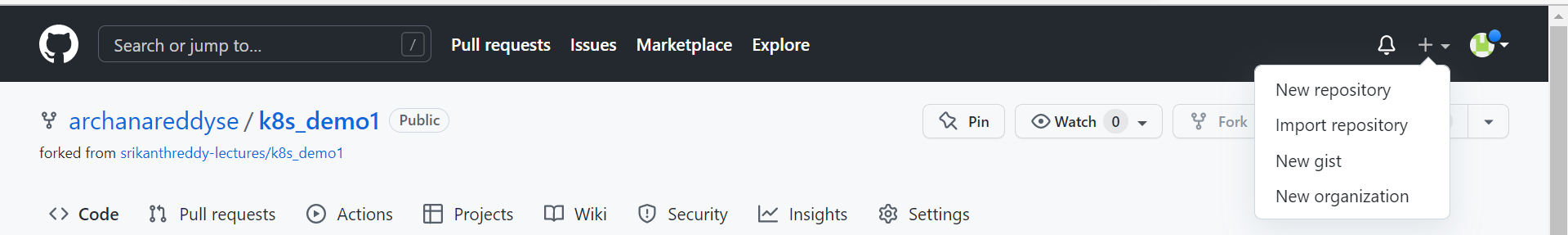
Description automatically generated

How will see how to local repository to git hub



Create the github account

Now create the new repository



Graphical user interface, text, application

Description automatically generated

Click on code will to get http link copy that and paste in git bash

Graphical user interface, text, application, email

Description automatically generated

$ git add .

$ git commit

$ git remote add origin copy the link of github repository

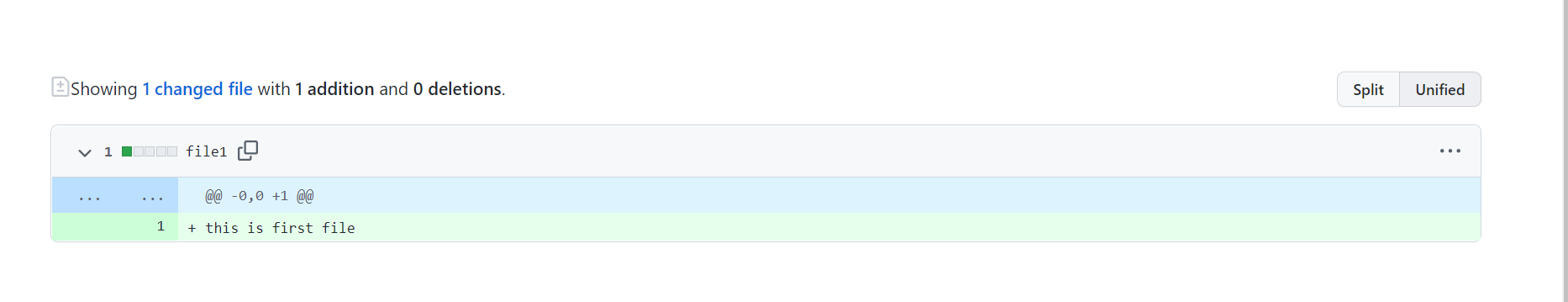
Text

Description automatically generated

Text

Description automatically generated

Now we check in git hub repository



Now will add few more files and change in file1 will see changes

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Graphical user interface, application

Description automatically generated

Branching Concepts:- when one or more persons are working on same project we use branching concept

We create file f1 in master and commit it as “a” commit

Text

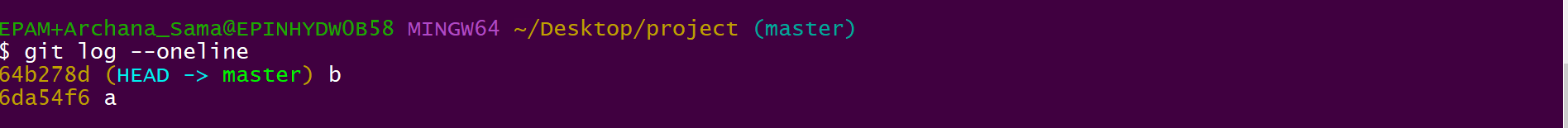
Description automatically generated

Create another file by f2 and commit as f2

Text

Description automatically generated

Check log by $git log –oneline , we can see “a & b” commit



Now we create branch as archana

$git branch <branch name>

We can see how many branches are present by

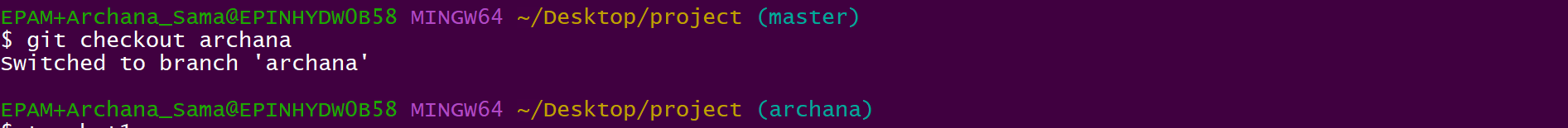
$ git branch {\* shows in which branch are in}

Text

Description automatically generated

To get into branch we go for

$git checkout <branch name>



Now create some files in archana branch as files t1 and commit as “c”

Text

Description automatically generated

Another file as t2 and commit as “d”

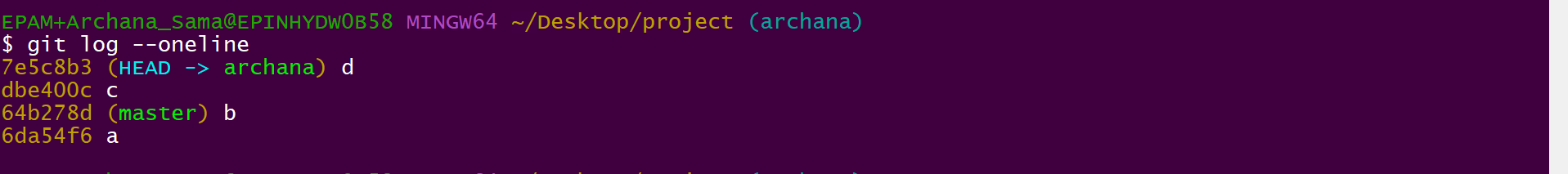
Text

Description automatically generated

Now check the log history we get all master branch and archana branch commit there

Commit -a,b are from master

-c,d are from archana

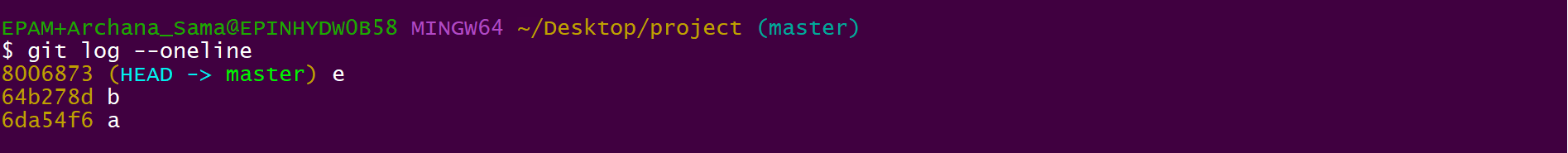


Now check out to master branch and create the another file f3 and commit as “e”

Text

Description automatically generated

Now check the log we see only commits of master branch only



Git merge concept

For merging we must be master branch

$git merge archana

A pop window will be opend ->:q

Text

Description automatically generated

Now check the log where will get all sequence of commits a,b,c,d,e and new commit a head

Text

Description automatically generated

<https://github.com/archanareddyse/kubernetes.git>

<https://github.com/archanareddyse/ansible.git>

<https://github.com/archanareddyse/docker.git>

<https://github.com/archanareddyse/Nagios.git>