**1. git config**

This command is used to configure an author name and email associated with your git activities.

* **Usage:**For configuring the author name globally.

**git config** –global user.name **[**name**]**

* **Usage:**For configuring the author name locally.

**git config** user.name **[**name**]**

* **Usage:** For configuring email addresses locally.

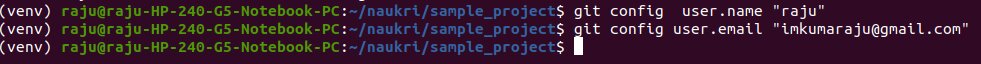
**git config** user.email **[**email address**]**

git config user.email [email address]

* **Usage:** For configuring email addresses globally.

**git config** –global user.email **[**email address**]**

**Example:**



**2. git init**

This command is used to initialize a new git repository.

* **Usage:** For initialising a new git repository.

**git init** **[**repository name**]**

Note: If you do not provide a name to the repository it defaults to *.git .*

**Example:**

2022_02_Git-commands-Git-Init-example.jpg

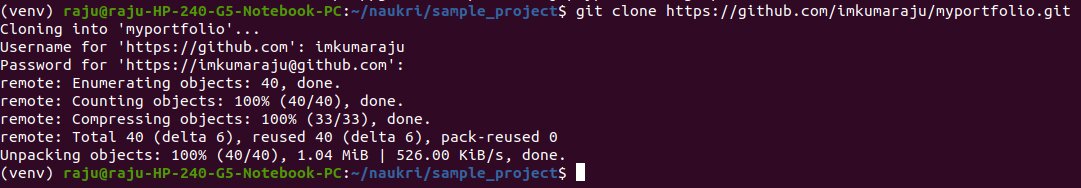
**3. git clone**

This command is used to clone a remote git repository.

* **Usage:** For cloning a git repository.

**git clone** **[**url**]**

**Example:**



**4. git add**

This command is used to add files to the staging.

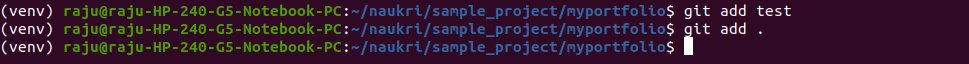
* **Usage:** For adding a particular file.

**git add** **[**filename**]**

* **Usage:** For add all files to staging.

**git add**

**Example:**



**5. git commit**

This command is used to record a file permanently in the project version history. It is a standard to add a message associated with the commit.

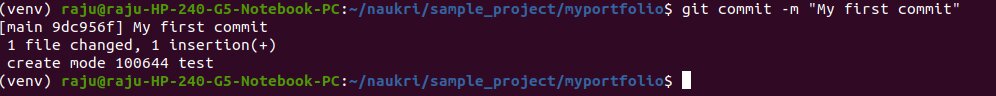
* **Usage:**For committing your staged changes.

**git commit** -m **[**message**]**

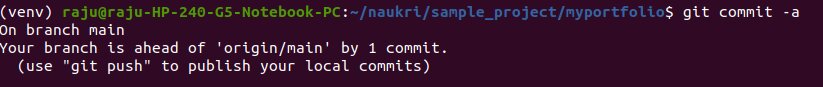
* **Usage:** For committing all the staged and unstaged changes till now. It is generally used when you have already added your file changes to the staging area using the *git add* command and need to add  additional file changes to the the staging area with the commit.

**git commit** -a

**Example 1:** For committing your staged changes.



**Example 2:** For committing both staged and unstaged files.



**6. git diff**

This command is used to check the current file changes.

* **Usage:** For checking all the unstaged file changes:

**git diff**

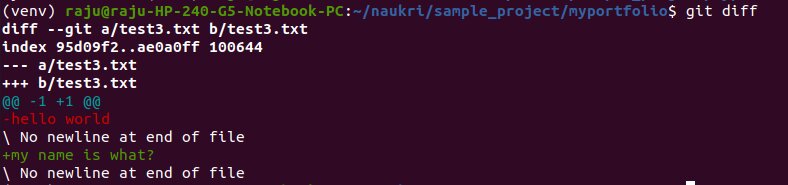
* **Usage:** For checking all the staged file changes:

**git diff** -staged

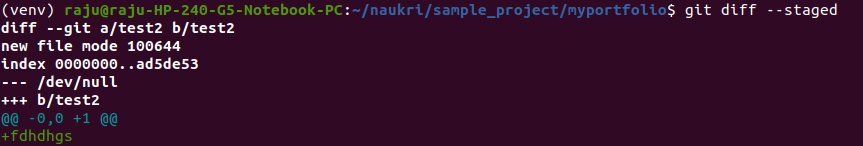
* **Usage:** For checking the files changes between two git branches:

**git diff** **[**first branch**]** **[**second branch**]**

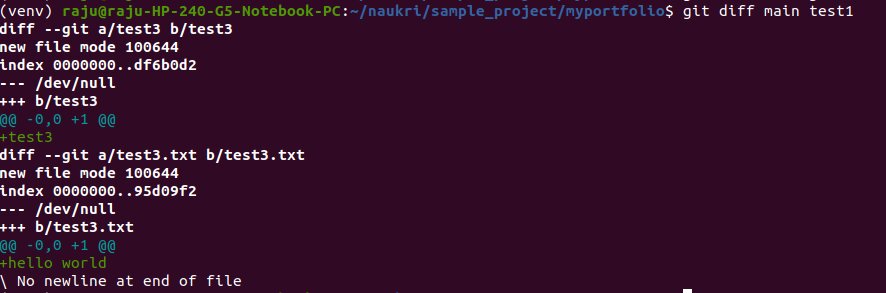
**Example 1:** Check all unstaged file changes.



**Example 2:**Check all the staged file changes.



**Example 3:**Check changes between two branches.



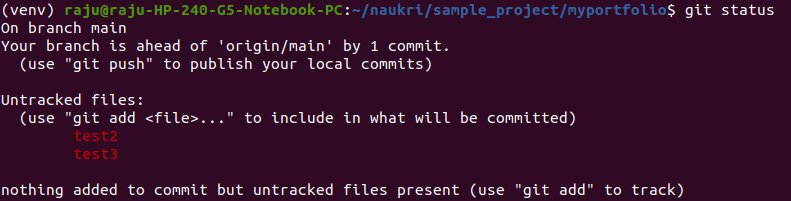
**7. git status**

This command is used to lists all the committed files.

* **Usage:** For listing all the files that have been committed:

**git status**

**Example:**



**8. git reset**

This command is used to unstage a file from the staging area.

* **Usage:** Unstage the files form staging area while keeping the file changes.

**git reset**

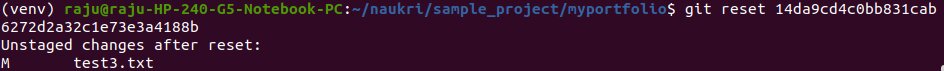
* **Usage:**Reset a commit.

**git reset** **[**commit **id]**

**Example 1:** Unstage the staged changes.

2022_02_Git-reset-command-example-1.jpg

**Example 2:** Resetting a commit.



**9. git rm**

This command is used to delete a specific file from the current working directory and stages the deletion.

* **Usage:** For deleting a specific file from the  current working directory and stages the deletion.

**git rm** **[**filename**]**

**Example:**Deleting the file *test3.txt*from the staged changes.

2022_02_Git-rm-command-example.jpg

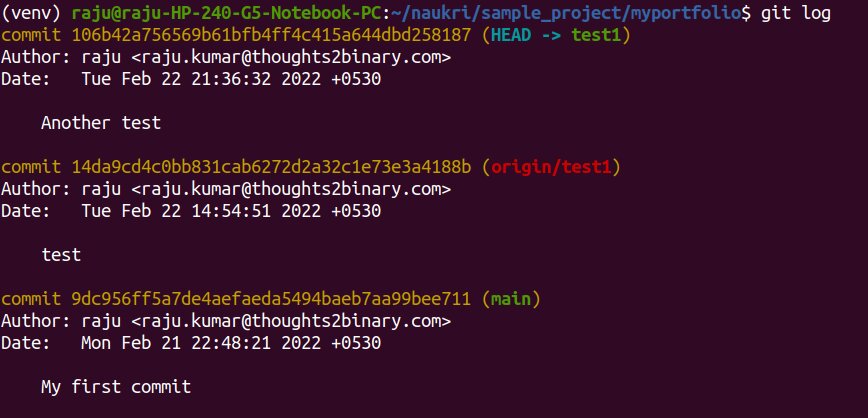
**10. git log**

This command is used for listing the version history of the current git branch.

* **Usage:** For listing the version history of the current branch:

**git log**

**Example:** Checking the version history of the current branch(i.e,test).



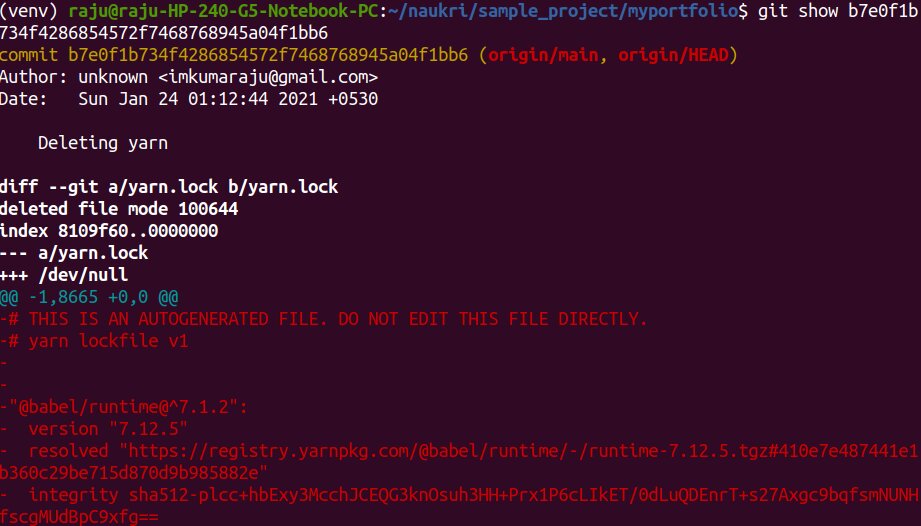
**11. git show**

This command is used to view the metadata and the file changes of a specific commit.

* **Usage:**Checking the metadata and file changes of a commit:

**git show** **[**commit **id]**

**Example:**Show the file changes and metadata of a commit.



**12.git tag**

This command is used to add a tag associated with a commit.

**Usage:** Adding a tag to a commit:

**git tag** **[**commit **id]**

**Example:**

2022_02_Git-tag-command-example.jpg

**13. git branch**

This command is used to create a branch from the current working directory.

* **Usage:**Creating a new branch:

**git branch** **[**branch name**]**

* **Usage:** For deleting the feature branch:

**git branch** -d **[**branch name**]**

**Example 1:** Creating a new git branch.

2022_02_Git-branch-command-example-1.jpg

**Example 2:** Deleting the feature branch.

2022_02_Git-branch-command-example-2.jpg

**14. git checkout**

This command is used for switching among different git branches.

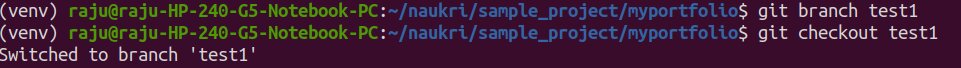
* **Usage:** Checkout a git branch:

**git checkout** **[**branch name**]**

* **Usage:**Create a new branch and switch into it:

**git checkout** -b **[**branch name**]**

**Example 1:**Checking out to an existing git branch.



**Example 2:**Create a new branch and checkout to it.

2022_02_Git-checkout-command-example-2.jpg

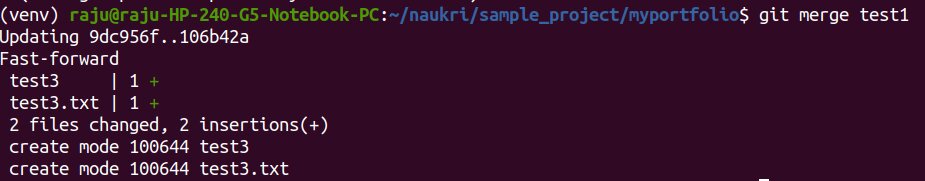
**15. git merge**

This command is used to merges the specified branch with the current branch.

**Usage:**Merging two branches:

**git merge** **[**branch name**]**

**Example:**



**16. git remote**

This command is used to connect the local git repository to the remote server.

* **Usage:**Connecting to the remote server:

**git remote** add **[**variable name**]** **[**Remote Server Link**]**

**Example:**

2022_02_Git-remote-command-example-2.jpg

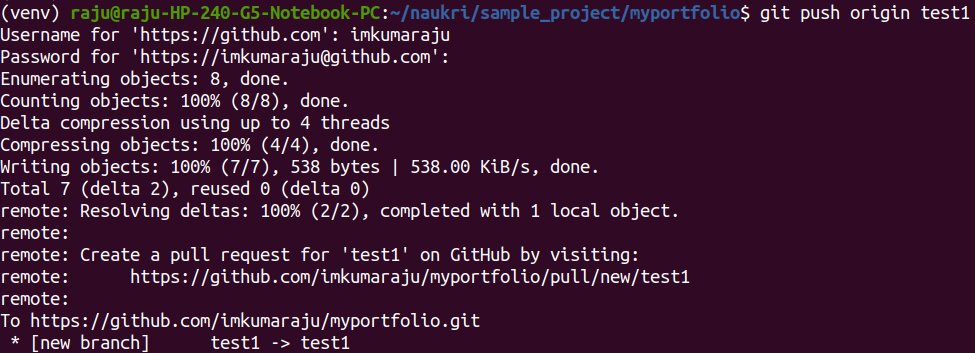
**17. git push**

This command is used to send your staged changes to the remote repository.

* **Usage:** Commit the staged changes to the remote repository.

**git push** **[**variable name**]** **[**remote repositry name**]**

**Example:**



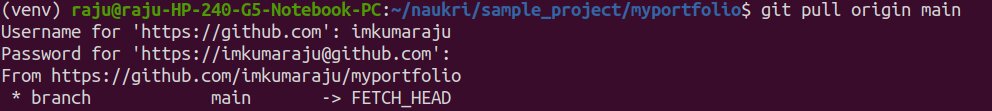
**18. git pull**

This command is used to get the changes in the remote repository and merge them to the current working directory.

* **Usage:** Pull changes from a remote repository:

**git pull** **[**variable name**]** **[**remote repositry name**]**

**Example:**



**19. git stash**

This command is used to temporarily store all the changed files in the working directory.

* **Usage:** Save all the modified tracked files temporarily:

**git stash**

* **Usage:** List all the stashes:

**git stash** list

* **Usage:**Delete the latest stash:

**git stash** drop

**Example 1:** Stashing the changes in the current working directory.

2022_02_Git-stash-command-example-1.jpg

**Example 2:** Listing all the stashes.

2022_02_Git-stash-command-example-2.jpg

**Example 3:**Discard the latest stash.

2022_02_Git-stash-command-example-3.jpg

**20. git fsck**

This command is used to check the integrity of the Git file system and it also helps in identifying the corrupted objects.

* **Usage:**Integrity check of git file system:

**git fsck**

**Example:**

