

MDM Div-A Exp No:01

Git CMDS

1. CREATE DIRECTORY AND INITIALIZE GIT

EXPLANATION: We start by creating a new directory for our project and initializing it as a Git repository.

```
amarrajendrapalwankar@fedora:/Exp01$ mkdir  
test
```

Creates a new directory named "test"

```
amarrajendrapalwankar@fedora:/Exp01$ cd test
```

Changes current directory to "test"

```
amarrajendrapalwankar@fedora:/Exp01/test$ vi  
test.txt
```

Opens vi editor to create and edit test.txt file - Add some content and save it

```
amarrajendrapalwankar@fedora:/Exp01/test$ ls
```

test.txt

Lists files in current directory - Confirms test.txt was created

```
amarrajendrapalwankar@fedora:/Exp01/test$ git init
```

Initializes an empty Git repository in current directory - Creates hidden .git folder

```
amarrajendrapalwankar@fedora:/Exp01/test$ git add test.txt
```

Stages test.txt file for commit - Prepares file to be tracked by Git

```
amarrajendrapalwankar@fedora:/Exp01/test$ git commit -m "ok"
```

Commits the staged file with message "ok" - Saves snapshot of your project

2. CHECK BRANCH AND CONFIGURATION

EXPLANATION: Verify which branch you are on and check your Git configuration settings.

```
amarrajendrapalwankar@fedora:/Exp01/test$  
git branch
```

* master

Shows current branch - The asterisk () indicates you are on "master" branch*

```
amarrajendrapalwankar@fedora:/Exp01/test$  
git config --list
```

user.email=amar.palwankar@famt.ac.in

user.name=Amar

Displays all Git configuration settings including your username and email

3. REMOVE CONFIGURATION (IF NEEDED)

EXPLANATION: If you want to remove existing user information from Git configuration.

If you want to remove above info

Remove email

`git config --global --unset user.email`

Removes the global email configuration

Remove name

`git config --global --unset user.name`

Removes the global username configuration

4. SET NEW CONFIGURATION (IF NOT SET)

EXPLANATION: Configure Git with your identity - This info appears in your commits.

If not set above then

```
git config --global user.name "Your Name"
```

Sets your name globally for all Git repositories on this system

```
git config --global user.email "your-email@example.com"
```

Sets your email globally for all Git repositories on this system

5. ADD REMOTE REPOSITORY

EXPLANATION: Connect your local repository to a remote repository on GitHub.

Need to be created first on GitHub

Go to GitHub.com and create a new repository named "mdm1" before running this command

```
git remote add origin https://github.com/amarfamt/mdm1.git
```

Links your local repo to remote GitHub repo - "origin" is the default name for remote

```
git branch
```

o/p: * master

Confirms you are still on master branch before pushing

6. PUSH TO REMOTE REPOSITORY

EXPLANATION: Upload your local commits to the remote GitHub repository.

```
git push origin master
```

Pushes your master branch to remote repository named "origin"

enter username: amarfamt and PAT (Token use classic token and select repo option)

GitHub requires Personal Access Token (PAT) instead of password - Generate it from GitHub Settings > Developer settings > Personal access tokens

7. PULL FROM REMOTE REPOSITORY

EXPLANATION: Download changes from remote repository to your local machine.

Now to pull repo in same local directory

`git pull origin master`

Fetches and merges changes from remote master branch to your local master branch

If contents are same then Already up to date.

This message means there are no new changes in the remote repository

8. TEST PULL WITH CHANGES

EXPLANATION: Test pulling changes by making modifications on GitHub first.

Make some changes in test.txt on repo and commit & then

Go to GitHub website, edit test.txt file directly, add some text and commit the changes

`git pull origin master`

Now this will download the changes you made on GitHub to your local repository

Now display content of test.txt from local using cat cmd

`cat test.txt`

Displays the content of test.txt - You will see the changes you made on GitHub are now in your local file