**Day9:Deep Dive in Git & GitHub for DevOps Engineers**

**1)What is Git and why is it important?**

Git is the most commonly used version control system. Git tracks the changes you make to files, so you have a record of what has been done, and you can revert to specific versions should you ever need to. Git also makes collaboration easier, allowing changes by multiple people to all be merged into one source. it is a distributed version control system that allows developers to track changes to their code and collaborate with others on the same codebase.

**Importance of GIt:**

Git is a speedy and efficient distributed [**VCS**](https://www.w3docs.com/learn-git/version-control-software.html) tool that can handle projects of any size, from small to very large ones. Git provides cheap local branching, convenient staging areas, and multiple workflows. It provides support for non-linear development. Git enables multiple developers or teams to work separately without having an impact on the work of others. It is important because it enables developers to work on the same codebase simultaneously without having to constantly send files back and forth or constantly merge changes manually.

**2)What is the difference Between Main Branch and Master Branch?**

**main** or **master** is default branch when you create a repository. GitHub uses **main** as its default branch while others systems still use **master.**

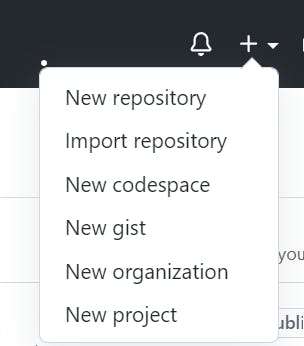
**3)Can you explain the difference between Git and GitHub?**

**Git:** Git is a version control system that allows developers to manage changes to their code. GIT will allow installing it on your local system.

**GitHub:** GitHub is a platform that provides hosting for Git repositories. With GitHub, developers can store their code remotely and collaborate with others on the same codebase. Additionally, GitHub provides a web-based interface for viewing and managing code, as well as several additional features such as issue tracking, pull requests, and wikis.

**4)How do you create a new repository on GitHub?**

1. Go to [github.com](http://github.com/) and log in to your account.
2. Click the “New repository” button.



1. Enter a name for your repository and add a brief description.
2. Choose whether you want the repository to be public or private.

Choose repository visibility. Click on the public if you want your repository to be cloned by anyone.

1. Click the “Create repository” button.

**5)What is the difference between local & remote repositories? How to connect local to remote?**

**The local repository** is a Git repository that is **stored on your computer.** Local repositories reside on the computers of team members.

**The remote repository** is a Git repository that is **stored on some remote computer.** In contrast, remote repositories are hosted on a server that is accessible for all team members - most likely on the internet.

**Tasks**

**1)Set your user name and email address, which will be associated with your commits.**

To set your user name and email address, which will be associated with your commits in Git, you can use the following command:

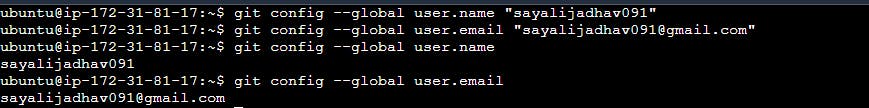
git config — global [user.name](http://user.name/) “<name>”

git config — global [user.name](http://user.name/) “<email id>”

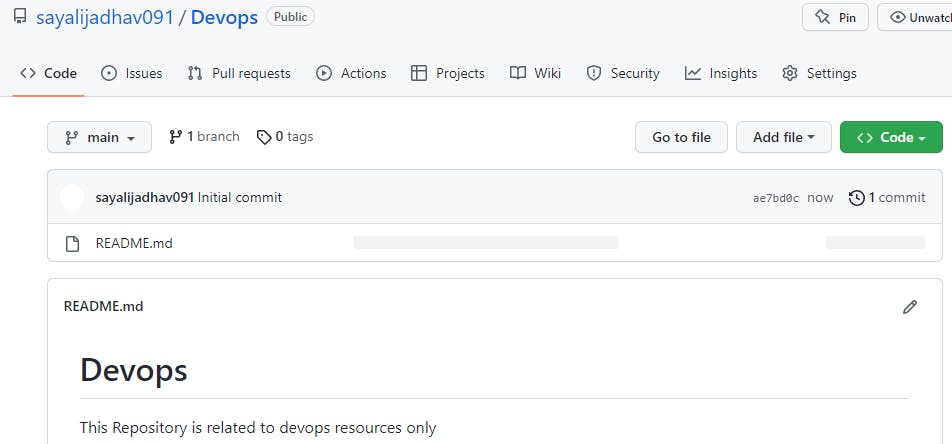
To check the details :

git config — global [user.name](http://user.name/)

git config — global [user.name](http://user.name/)

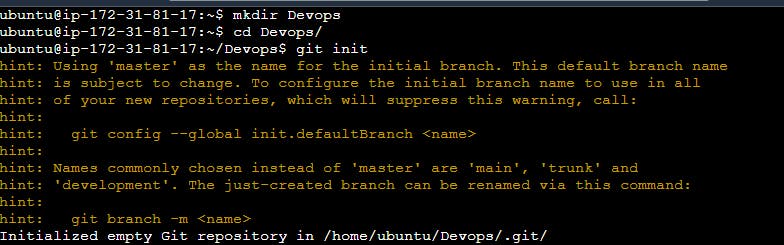


**2)Create a repository named “Devops” on GitHub**



**3)Connect your local repository to the repository on GitHub.**

To connect your local repository to the repository on GitHub, you’ll first need to initialize a new Git repository on your local machine. To do this, navigate to the directory where you want to create your local repository, and then use the command “**git init**”.

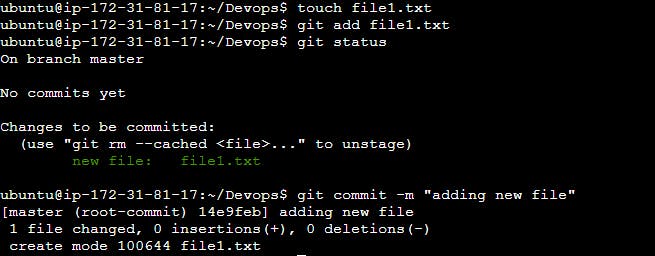


Add the files in your new local repository. This stages them for the first commit.

**git add**

Commit the files that you've staged in your local repository.

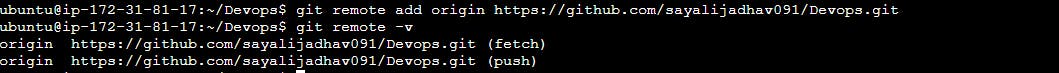
**git commit -m "adding new file"**



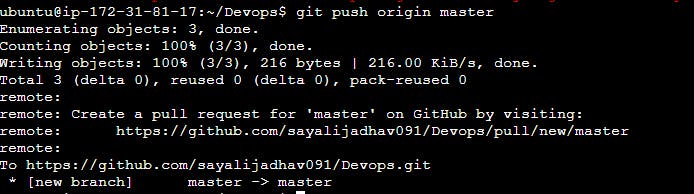
Copy remote repository URL field from your GitHub repository, in the right sidebar, copy the remote repository URL.

In Terminal, add the URL for the remote repository where your local repository will be pushed.

**git remote add origin <remote repository URL>**



Push the changes in your local repository to GitHub. use the command “**git push origin master**” to push your local commits to the remote repository on GitHub.



Now we have successfully Connected your local repository to the repository on GitHub.