GitHub Project - Getting Started with Git - Creating a Local Repository

git init command is used for creating a new blank repository or make an existing project as a git project. This command creates a .git folder that contains subdirectory and a HEAD file that stores the HEAD of the master branch.

**INSTRUCTIONS**

* Create a devops directory in your home directory
* Go to devops directory
* Run git init to initialize a git repository

GitHub Project - Getting Started with Git - Setup Your username and email in the Git repository

Let's configure our email and name in the git repository. When we do commit, GIT will associate that commit to our name and email so that other developers in the team can identify the committer.

We use git config command for the same

**INSTRUCTIONS**

* Make sure you are in ~/devops directory
* Specify your name with the below command. Replace full\_name with your full name like Abhinav Singh
* git config user.name "full\_name"
* Specify your email with the below command. Replace email\_id with your email id like abc@xyz.com

git config user.email "email\_id"

GitHub Project - Getting Started with Git - Create a File in Your Repository

Let's create a file in the repository.

**INSTRUCTIONS**

* Make sure you are in ~/devops directory
  + Create an empty file called helloworld.py using touch helloworld.py
  + Add line print("Hello World") in helloworld.py and save the file
  + Please use the below command to add a line to the file

echo 'print("Hello World")' > helloworld.py

* + Verify the file content using cat command

cat helloworld.py

* + You can use nano or vi editor to edit the file content if you are comfortable with them using commands like nano helloworld.py or vi helloworld.py

GitHub Project - Getting Started with Git - Make your first Commit

Commit is used to snapshot the changes made to the files in a git repository. First, the changed files are added to the staging area using git add command

And then we commit them to a branch using git commit command. Please note that we provide a message to git commit using -m flag so that other developers can understand the gist of commit.

Check the branch you are currently using the git branch command. You should be on the master branch. Commit your changes to the master branch using the git commit command.

**INSTRUCTIONS**

* Make sure you are in ~/devopsdirectoy
* Add the file helloworld.py to the staging area using the git add helloworld.py command.
* Commit the changes made using git commit with the message "First commit". Use the below command

git commit -m "First commit"

GitHub Project - Getting Started with Git - Creating a Branch

We can create a new branch using the command git branch <branch\_name>.

For example, to create a testing branch, type below command in the terminal

git branch testing

To checkout to testing branching, type below command

git checkout testing

With -b switch we can combine the above two steps into one. For example, to create the dev branch and checkout to it at the same time, user below command

git checkout -b dev

**INSTRUCTIONS**

* Make sure you are in ~/devops directory
* Check the branch you are in using git branch command
* If you are not in master then checkout to master branch using git checkout master command
* Now create a new branch feature\_a and checkout to that branch

GitHub Project - Getting Started with Git - Making changes in a Branch

When you create a new branch it isolates your work from the point you created the branch. Now you can work on the new branch separately without affecting the original branch. This can help you to isolate your work from other developers working on the same project.

**INSTRUCTIONS**

* Make sure you are in ~/devops directory
* Check the branch you are in using git branch command
* If you are not on feature\_a branch then checkout to feature\_a branch using git checkout feature\_a command
* Make sure the first line is print("Hello World") in file helloworld.py
* Append the line print("I am learning git") to the file helloworld.py. We can use either vi or nano editor to append the line. Or you can use echo command.
* Check the status of files in your project using git status command

GitHub Project - Getting Started with Git - Commit in the new branch

You can make commits in the new branch. This will take snapshot of the changes you made in that branch.

**INSTRUCTIONS**

* Make sure you are in ~/devops directory
* Check the branch you are in using git branch command
* If you are not on feature\_a branch then checkout to feature\_a branch using git checkout feature\_a command
* Check the working tree status using git status command
* helloworld.py file should be shown as modified under **Changes not staged for commit:** when git status command is run
* Add the helloworld.py file to staging area using git add command
* Commit the change to branch feature\_a using git commit command with the message "Added I am learning git line"
* -m flag can be used to provide message while committing

GitHub Project - Getting Started with Git - Merging branches

You can merge the changes you made in one branch into another branch. You also need to resolve any conflicts if they arise.

**INSTRUCTIONS**

* Make sure you are in ~/devops directory
* If you are not on the master branch then checkout to master branch using git checkout master command
* Check the content of helloworld.py file using cat helloworld.py command
* Merge feature\_a branch into master branch using git merge <branch\_name> command
* Check the content of helloworld.py file again

GitHub Project - Getting Started with Git - Create a repository on GitHub

Signup or sign in to your GitHub profile. Create a new repository with your preferred name. This will create an empty repository.

**INSTRUCTIONS**

* Sign in to your GitHub account
* If you don't have a GitHub account create one
* Create a repository with your preferred name

GitHub Project - Getting Started with Git - Add remote to existing repository

You can add remote using git remote add origin command.

**INSTRUCTIONS**

* Make sure you are in ~/devops directory
* Add remote to your existing repository using : git remote add origin git@github.com:<your-github-username>/<repository-name>.git

**Note**: Please use ssh url of your repository instead of https url to be able to push the code.

GitHub Project - Getting Started with Git - Creating SSH keys and adding to GitHub

**What are SSH keys?**

SSH keys are used to identify yourself to GitHub without using username and password every time. SSH keys come in pair, a public key which is used by services like GitHub, and a private key which is stored only on our computer.

**How to generate SSH keys?**

Generate your ssh key pair using ssh-keygen -t rsa -b 4096 It will ask where you want to save the key. The default location is ~/.ssh/id\_rsa. Press enter to continue. After this, it will ask for the passphrase. You can give an empty password by pressing enter. You will get a confirmation showing the file location.

The private key is stored in ~/.ssh/id\_rsa file and public key is stored in ~/.ssh/id\_rsa.pub file. You can see the content of your public key using cat ~/.ssh/id\_rsa.pub command. Copy the output.

**How to add SSH keys to your GitHub account?**

Login to your GitHub account. Go to GitHub Settings by clicking on your display profile and select SSH and GPG keys from the side menu. Click on Add SSH keys button. Give any name to your key and paste the copied key.

Never share your private key. The public key can be shared anywhere.

**Note:**

* The .ssh folder and key files should have correct permissions. You can check permissions using ls -la command and change them using chmod command
* The .ssh directory permissions should be 700 (drwx------)
* The public key (.pub file) permissions should be 644 (-rw-r--r--)
* The private key (id\_rsa) permissions should be 600 (-rw-------)

GitHub Project - Getting Started with Git - Push your repository to remote server

You can push your repository to remote using git push command

**Note:**

* To push a branch on remote, your branch needs to have the latest changes present in remote repository
* If you get the failed to push error, first do git pull the branch to get the latest commits and then push it

**INSTRUCTIONS**

* Make sure you are in ~/devops directory
* Push the master branch to the remote repository using git push -u origin master command

GitHub Project - Getting Started with Git - Cloning a Repository

You can clone any repository you have access to using the git clone command. This will create a local copy of the repository.

git clone https://github.com/singh-ashok25/webserver.git

This will clone the repository in the same directory you run the command.

**INSTRUCTIONS**

* Go to your home directory using cd ~
* Clone this public repository https://github.com/singh-ashok25/webserver using git clone command