**Topic-GIt**

===================

**What is Git?**

**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.**

**Ways to Use Git**

**Git is software that you can access via a command line (terminal), or a desktop app that has a GUI (graphical user interface) such as Source tree shown below.**

**Git Repositories**

**======================**

**A Git repository is the . git/ folder inside a project. This repository tracks all changes made to files in your project, building a history over time. Meaning, if you delete the**

**. git/ folder, then you delete your project's history.**

**What are GitHub repositories used for?**

**GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.**

**Why you Using Git ?**

**Git is a free and open source version control. The best thing about open source software (like Git) is arguably freedom. Free as in speech (software that is free for all to use and modify). ...**

**Who runs GitHub?**

**Microsoft**

**As of November 2021, GitHub reports having over 73 million developers and more than 200 million repositories (including at least 28 million public repositories). It is the largest source code host as of November 2021.**

**...**

**GitHub.**

**GitHub's logo**

**Type of business Subsidiary**

**Parent Microsoft**

**URL github.com**

**Git Repositories:**

**They have 3 stages we have 1 working are**

**2 Staging area**

**3 Local Repo**

**Working area:**

**The working area is where files that are not handled by git. These files are also referred to as "untracked files." Staging area is files that are going to be a part of the next commit, which lets git know what changes in the file are going to occur for the next commit. ... They can also be called untracked files.**

**Staging area:**

**The staging area is like a rough draft space, it's where you can git add the version of a file or multiple files that you want to save in your next commit (in other words in the next version of your project).**

**Local Repo:**

**The local repository is a Git repository that is stored on your computer. The remote repository is a Git repository that is stored on some remote computer. ... When you are finished with doing changes into your workspace, you can add them to staging area and from there you can commit the changes to your local repository.**

**Lab work**

**Frist we can install the git in your system**

**Linux: yum install git -y**

**Ubuntu: sudo apt install git**

**Gitbash download in you system**

**Once we install git in your system and go create GitHub account in the below**

**url:** [**https://github.com/**](https://github.com/)

**Here what we give the GitHub username and password save it**

**After go your Linux system**

**Go to the root user and check the version Git**

**[root@ip-172-31-85-184 ~]# git --version**

**git version 2.32.0**

**[root@ip-172-31-85-184 ~]#**

**[root@ip-172-31-85-184 ~]#**

**[root@ip-172-31-85-184 ~]# git config –list =====it will display the list of username and mailid**

**[root@ip-172-31-85-184 ~]#**

**We need add the user name and mail id in git because we need to the connecting the GitHub**

**[root@ip-172-31-85-184 ~]#**

**[root@ip-172-31-85-184 ~]# git config --global user.name "sivakumar"**

**[root@ip-172-31-85-184 ~]# git config --global user.email** [**sivakumar.cheri21@gmai.com**](mailto:sivakumar.cheri21@gmai.com)

**[root@ip-172-31-85-184 ~]# git config --list**

**user.name=sivakumar**

**user.email=sivakumar.cheri21@gmai.com**

**[root@ip-172-31-85-184 ~]#**

**Here we generate the secure file ssh**

**[root@ip-172-31-85-184 ~]# ssh-keygen**

**Generating public/private rsa key pair.**

**Enter file in which to save the key (/root/.ssh/id\_rsa): type enter**

**Enter passphrase (empty for no passphrase): type enter**

**Enter same passphrase again: type enter**

**Your identification has been saved in /root/.ssh/id\_rsa.**

**Your public key has been saved in /root/.ssh/id\_rsa.pub.**

**The key fingerprint is:**

**SHA256:MU4OdavQf0AAeC58CfcPzpnidd8M8CFoX1iYZaT/nNI root@ip-172-31-85-184.ec2.internal**

**The key's randomart image is:**

**+---[RSA 2048]----+**

**| ...o.o=+ |**

**| o oo oo+. |**

**| . =oo=.+o |**

**| o +\*==+oo |**

**| o +S\*.=o. |**

**| . \* +.o+ . |**

**| . o . ..+E |**

**| . ..o |**

**| |**

**+----[SHA256]-----+**

**[root@ip-172-31-85-184 ~]#**

**[root@ip-172-31-85-184 ~]# cd .ssh**

**[root@ip-172-31-85-184 .ssh]# ls**

**authorized\_keys id\_rsa id\_rsa.pub**

**[root@ip-172-31-85-184 .ssh]#**

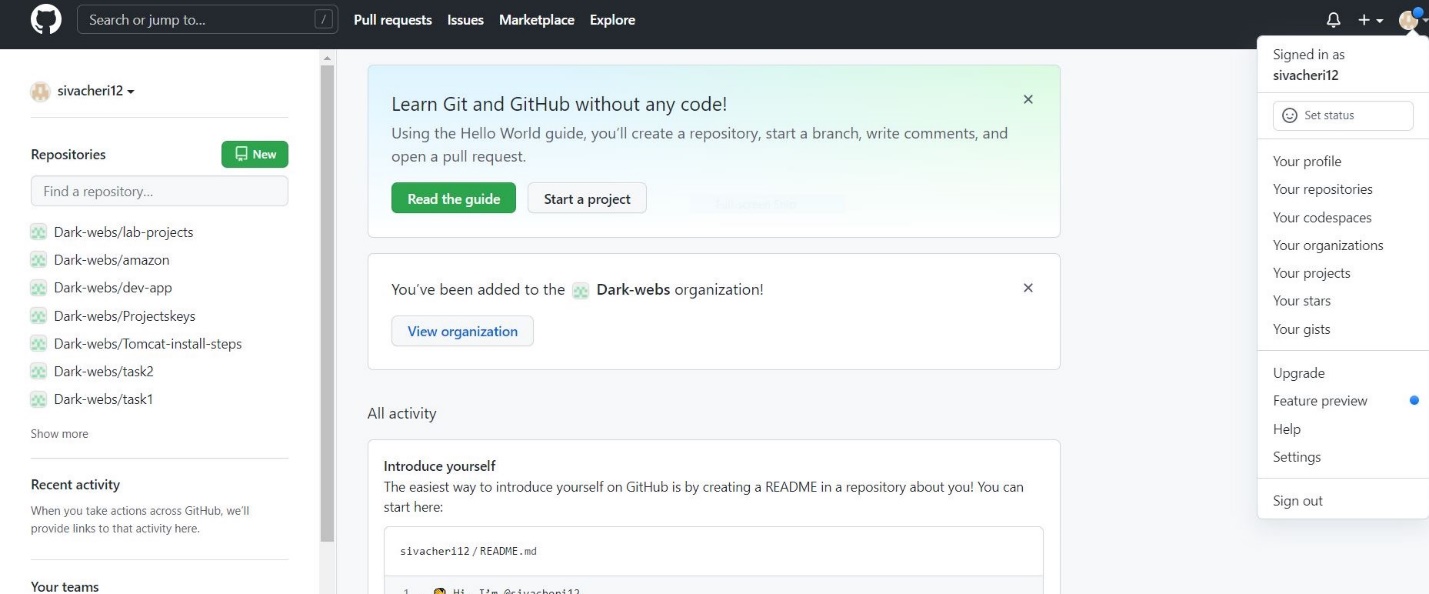
**[root@ip-172-31-85-184 .ssh]# cat id\_rsa.pubssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDKAp7DXyllgjef5Fn7j60Zlswv2PW3813HVCnwD3BucYDwmAY5xpPPut8dSvY+AiI3de0XGrzgMe9Sd89z3LuOKZvA/PEhmz9jIJ6ucdxGE/zvkFdA/1Mu3GyMCJNSJ7Kv68xEYNLouWWPOqZE/9FkGs7g9fs92P8BE1EhN6n1hGoGQq6NF3R7E7+BcSXQbn0S7C/ptf2jGftk31bJ9p+wqWkdI03yU/YQCEl9TZ3YkSCfDnFPZaz/EkLtLQqaATpaiKyNLz7cvu5Paketp2vcr4lU698Py92nJtzVkDis4luHoiPH8xNKszbAqiWRL+WSb9yBUzdzKeh3U5srl+kt root@ip-172-31-85-184.ec2.internal**

**Here we copy the SSH key go to the your github setting go the Right side table we see one options SSH and GPG Keys click that one it will show you two options 1 ssh keys 2 gpg key**

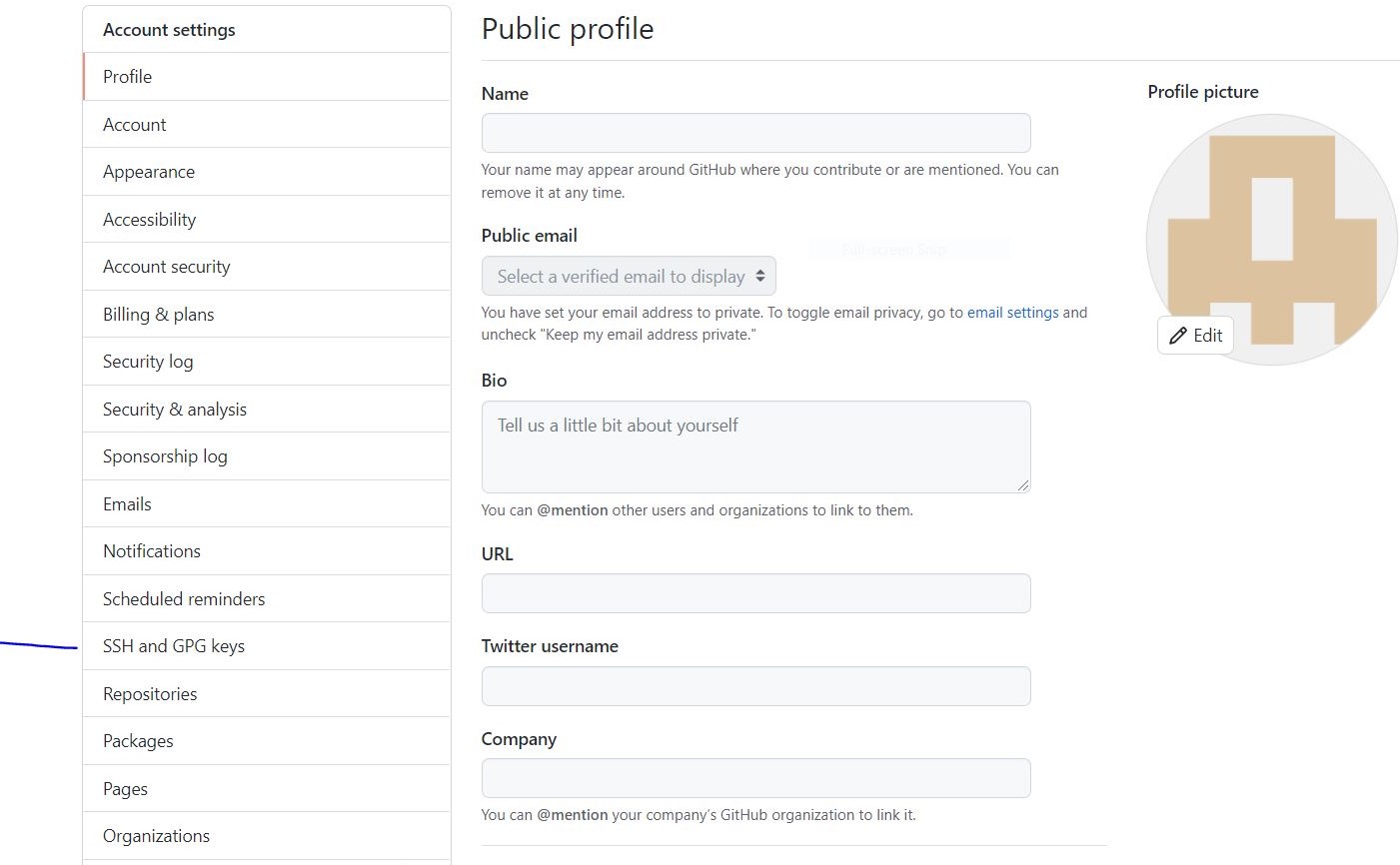
**We providing SSH KEYS here we can one green color New SSH KEY click on that one**

**It is display Title we enter the name SSH Key**

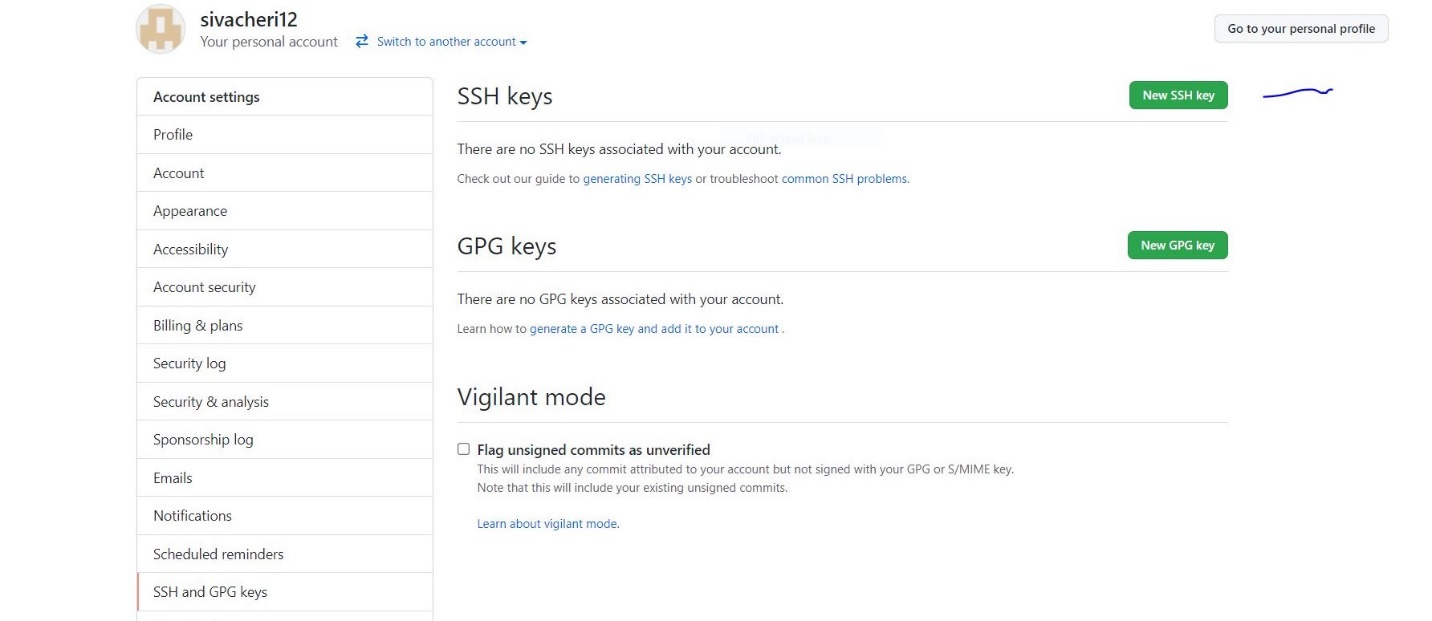
**And down side whatever your copy pervious that ssh key paste here in that box click the green color ADD SSH KEY**

****

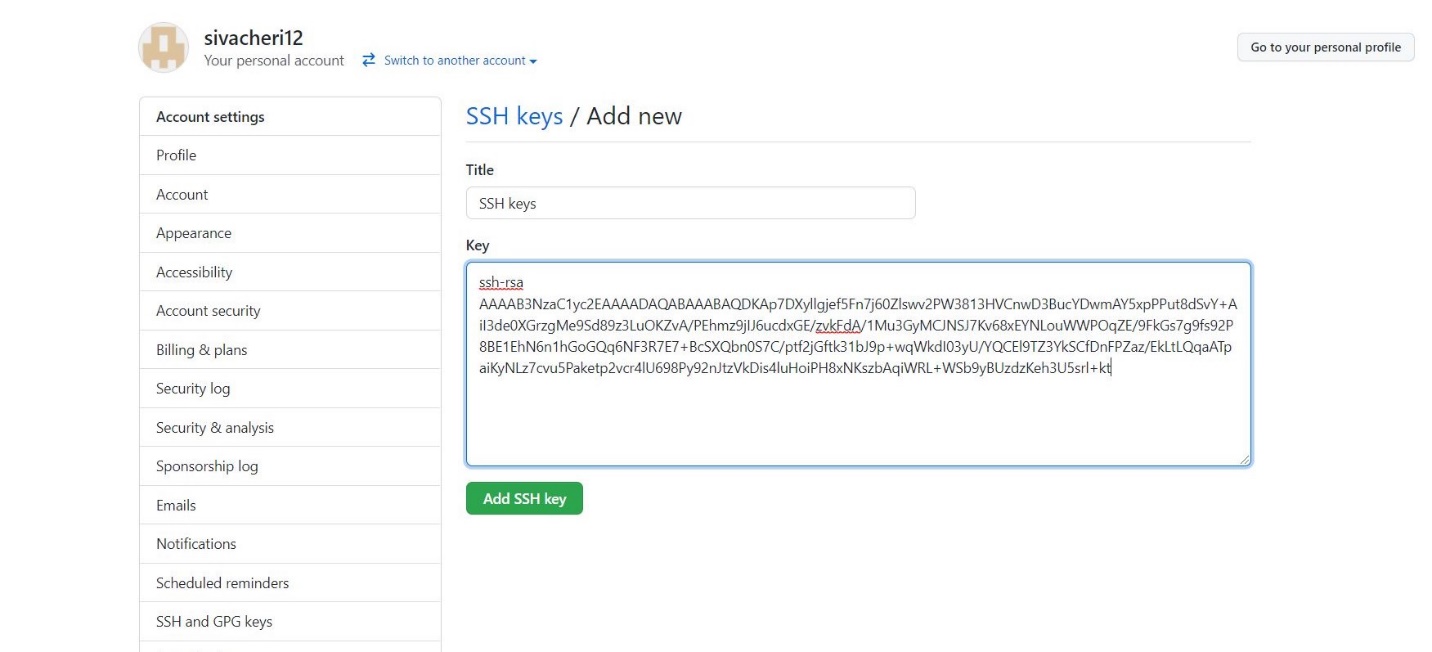
Click the ssh and GPG KEYS



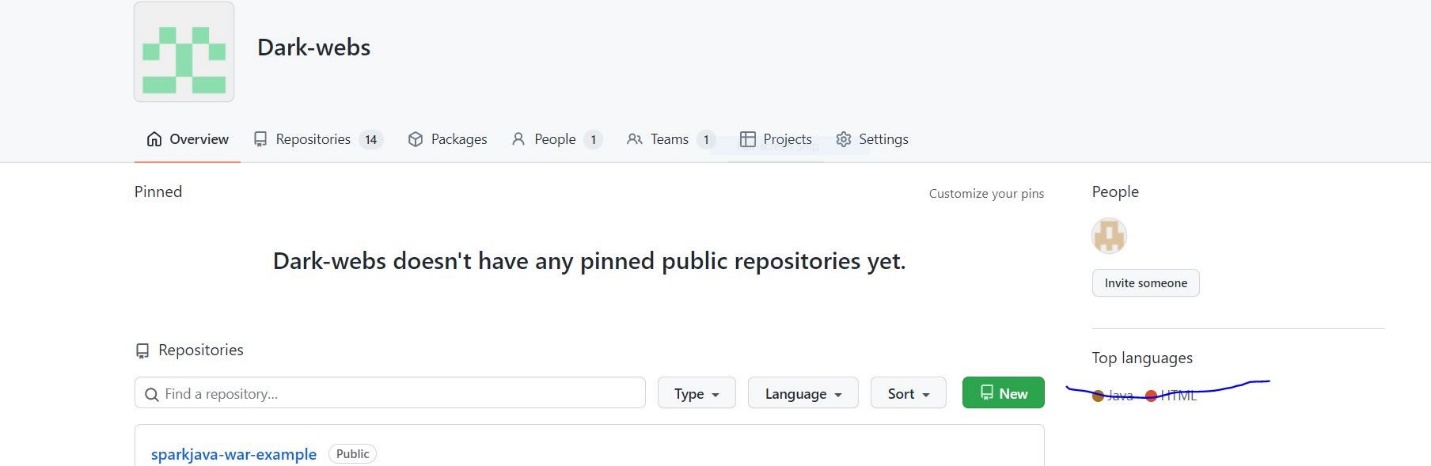
Enter the New SSH key



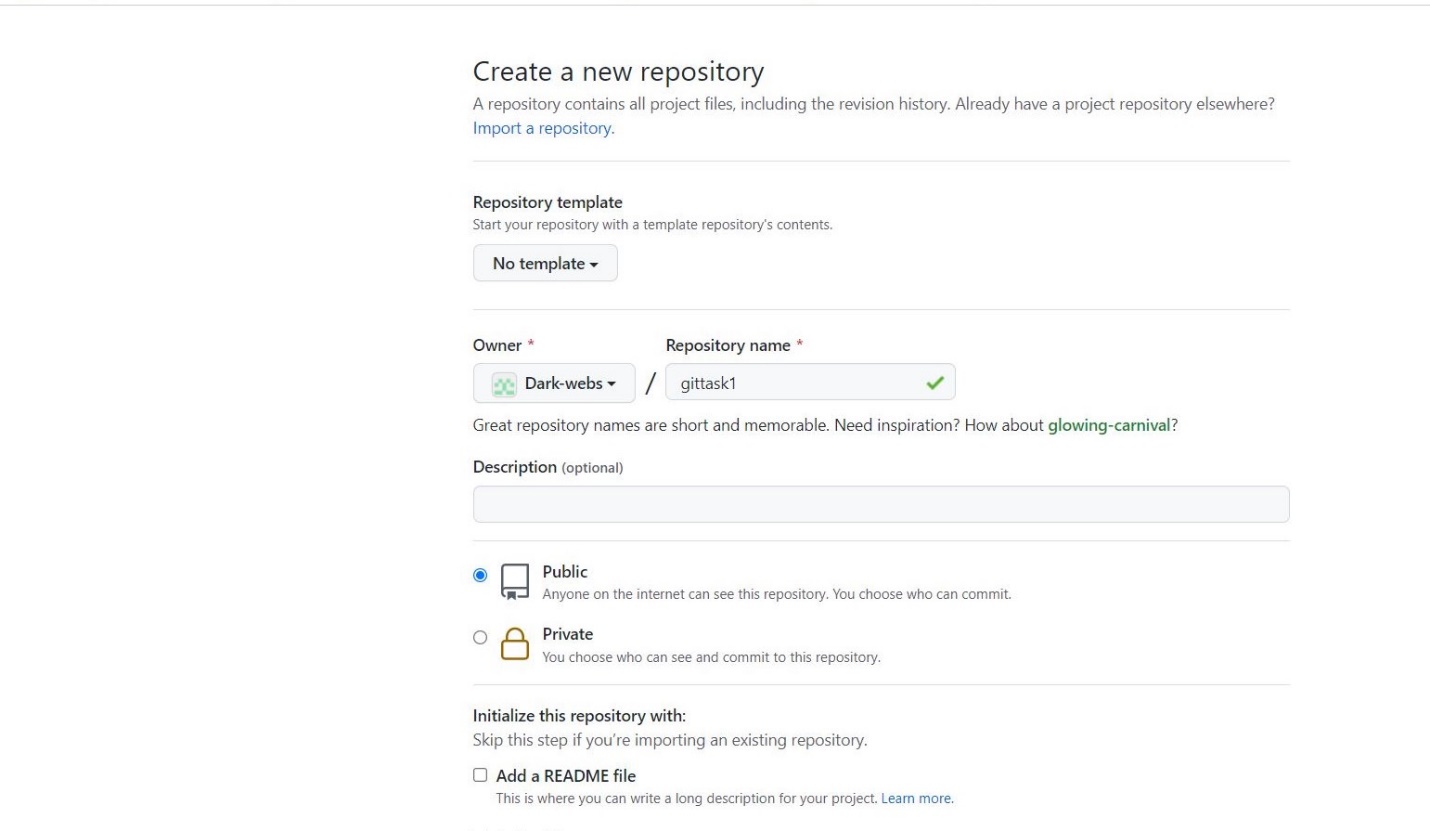
**Here give the SSH Key and the ADD SSH Keys**

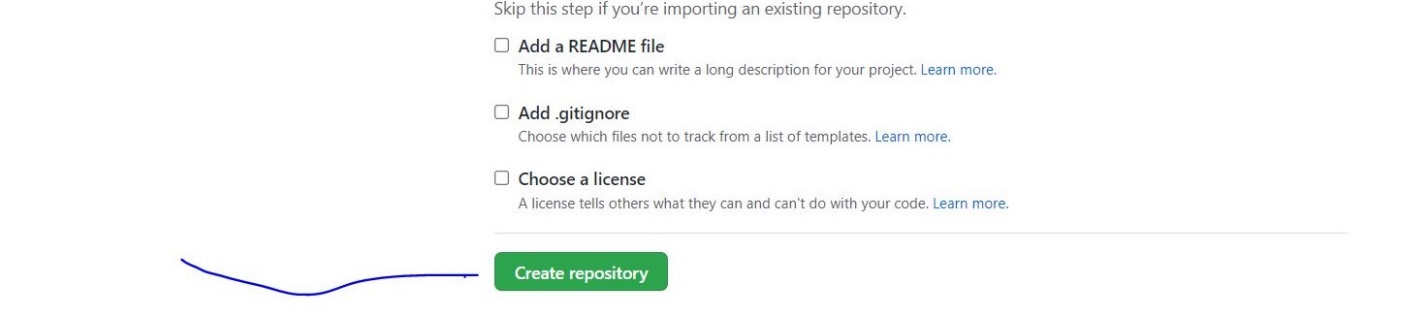
****

**Here we create one repository go to the GitHub click the New it will go to the new repository**

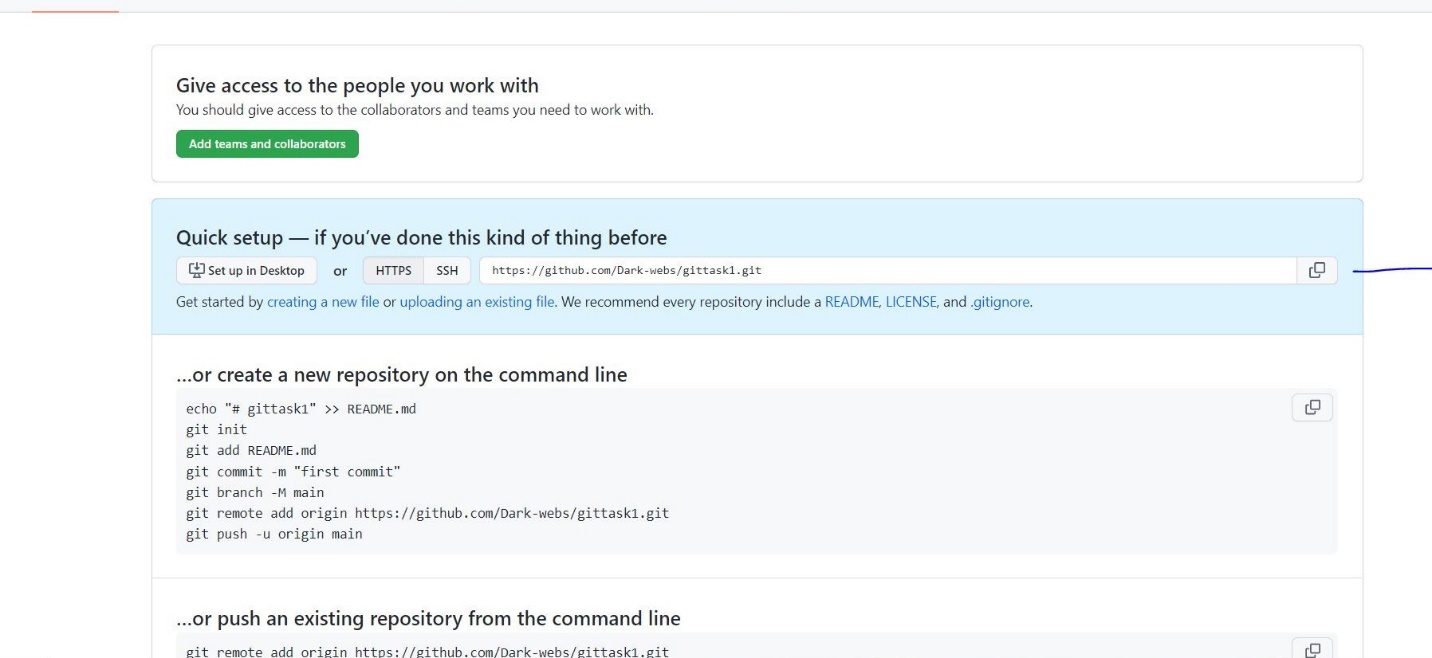
****

**Here we giving the name of repository and selecting the public and click the create repository**

****

****

**Copy the url paste your local remote**

****

**Go to the your git machine here creating file to uploading the remote repository**

**Please follow the steps**

**Frist we installations the git init**

**Git init we create local the local repository**

**So first we check ls –la it will display hiden files**

[root@ip-172-31-85-184 task1]# git init

hint: Using 'master' as the name for the initial branch. This default branch name

hint: is subject to change. To configure the initial branch name to use in all

hint: of your new repositories, which will suppress this warning, call:

hint:

hint: git config --global init.defaultBranch <name>

hint:

hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and

hint: 'development'. The just-created branch can be renamed via this command:

hint:

hint: git branch -m <name>

Initialized empty Git repository in /root/task1/.git/

[[root@ip-172-31-85-184 ~]# ls -ltra

total 174952

-rw-r--r-- 1 root root 169983496 Mar 15 2017 jdk-8u131-linux-x64.rpm

-rw-r--r-- 1 root root 129 Oct 18 2017 .tcshrc

-rw-r--r-- 1 root root 100 Oct 18 2017 .cshrc

-rw-r--r-- 1 root root 176 Oct 18 2017 .bashrc

-rw-r--r-- 1 root root 18 Oct 18 2017 .bash\_logout

drwxr-xr-x 6 root root 99 Nov 14 09:12 apache-maven-3.8.4

-rw-r--r-- 1 root root 9130223 Nov 14 13:25 apache-maven-3.8.4-bin.zip

dr-xr-xr-x 18 root root 257 Jan 24 05:14 ..

-rw------- 1 root root 1537 Jan 24 06:05 .viminfo

-rw-r--r-- 1 root root 250 Jan 24 06:05 .bash\_profile

drwxr-xr-x 3 root root 24 Jan 24 06:06 .m2

drwxr-xr-x 2 root root 40 Jan 24 06:20 .oracle\_jre\_usage

drwxr-xr-x 5 root root 93 Jan 24 06:21 sparkjava-war-example

-rw-r--r-- 1 root root 61 Jan 24 08:38 .gitconfig

drwx------ 2 root root 61 Jan 24 08:45 .ssh

-rw------- 1 root root 2215 Jan 24 09:19 .bash\_history

drwxr-xr-x 7 root root 119 Jan 24 10:19 .git

dr-xr-x--- 8 root root 326 Jan 24 10:19 .

**Now we creating the one directory in that directory we putting some files after we upload**

**Local repository to remote repository**

**[root@ip-172-31-85-184 ~]# mkdir task1**

**[root@ip-172-31-85-184 ~]# cd task1/**

**[root@ip-172-31-85-184 task1]# touch java**

**[root@ip-172-31-85-184 task1]# touch mysql**

**[root@ip-172-31-85-184 task1]# touch data**

**[root@ip-172-31-85-184 task1]# ls**

**data java mysql**

**[root@ip-172-31-85-184 task1]#**

**After we can check the status It will display the working area**

**[root@ip-172-31-85-184 task1]# git status**

**On branch master**

**No commits yet**

**Untracked files:**

**(use "git add <file>..." to include in what will be committed)**

**data**

**java working area**

**mysql**

**Nothing added to commit but untracked files present (use "git add" to track)**

**[root@ip-172-31-85-184 task1]#**

**Now im going to add working area to staging area**

[root@ip-172-31-85-184 task1]# **git add data**

[root@ip-172-31-85-184 task1]# **git status**

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: data

Untracked files:

(use "git add <file>..." to include in what will be committed)

java

mysql

[root@ip-172-31-85-184 task1]#

**Now I am going to commit staging area to local to area**

[root@ip-172-31-85-184 task1]# **git commit -m "task updated**"

[master (root-commit) 3fb7ea6] task updated

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 data

[root@ip-172-31-85-184 task1]#

**Now I’m checking remote repository**

[root@ip-172-31-85-184 task1]# git remote -v

[root@ip-172-31-85-184 task1]#

**Here it will display the remote repository so we to add repository go the your repository copy your**

**Remote url https or ssh any one we can copy go to the local machine and paste git remote add origin**

**https://github.com/Dark-webs/gittask1.git**

[root@ip-172-31-85-184 task1]# git remote -v

[root@ip-172-31-85-184 task1]# git remote add origin https://github.com/Dark-webs/gittask1.git[root@ip-172-31-85-184 task1]# git remote –v

origin https://github.com/Dark-webs/gittask1.git (fetch)

origin https://github.com/Dark-webs/gittask1.git (push)

[root@ip-172-31-85-184 task1]#

Now I’m going the push the my file localto remote repository

**[root@ip-172-31-85-184 task1]# git push origin master**

**The authenticity of host 'github.com (140.82.112.3)' can't be established.**

**ECDSA key fingerprint is SHA256:p2QAMXNIC1TJYWeIOttrVc98/R1BUFWu3/LiyKgUfQM.**

**ECDSA key fingerprint is MD5:7b:99:81:1e:4c:91:a5:0d:5a:2e:2e:80:13:3f:24:ca.**

**Are you sure you want to continue connecting (yes/no)? yes**

**Warning: Permanently added 'github.com,140.82.112.3' (ECDSA) to the list of known hosts.**

**Enumerating objects: 3, done.**

**Counting objects: 100% (3/3), done.**

**Writing objects: 100% (3/3), 207 bytes | 207.00 KiB/s, done.**

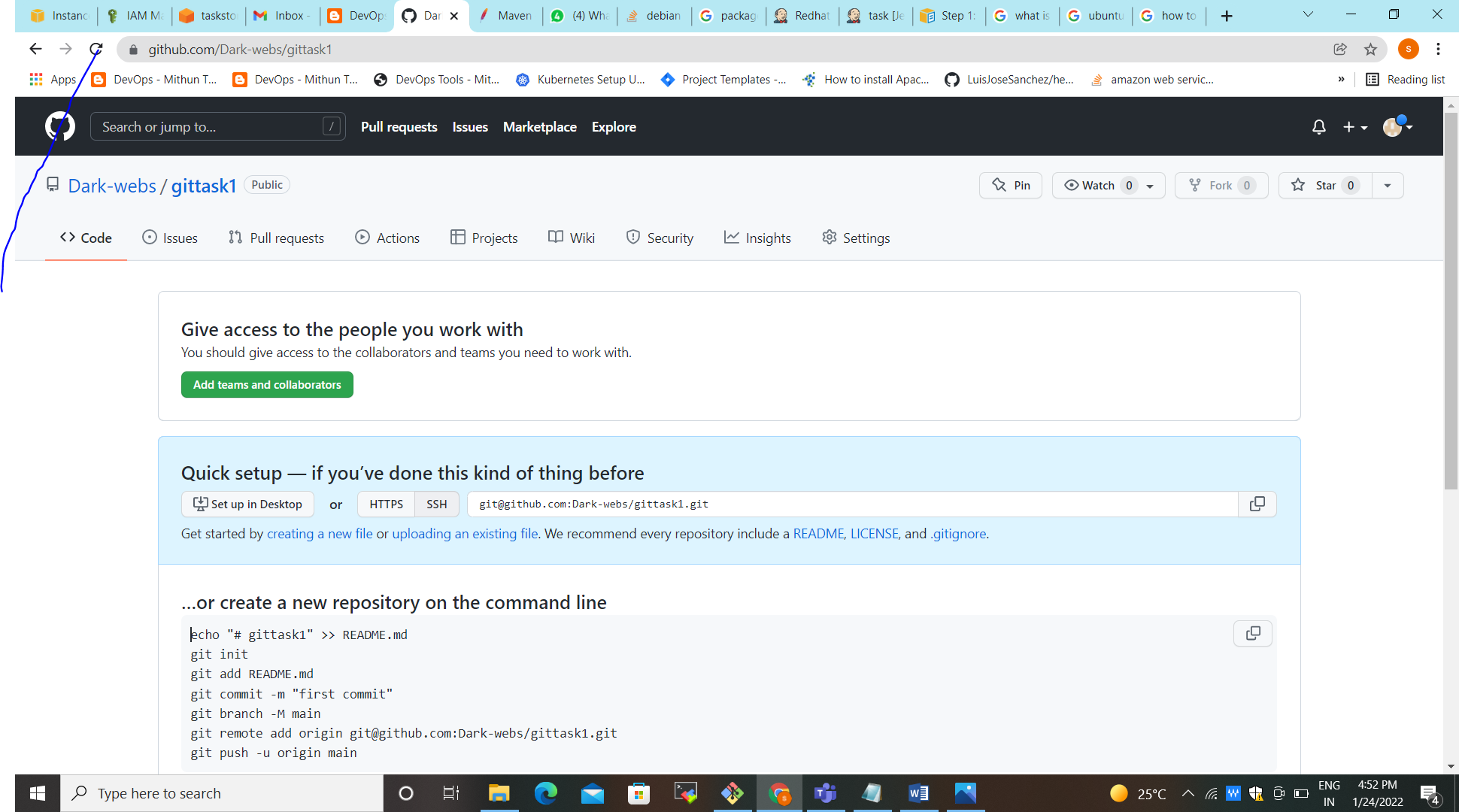
**Total 3 (delta 0), reused 0 (delta 0), pack-reused 0**

**To github.com:Dark-webs/gittask1.git**

**\* [new branch] master -> master**

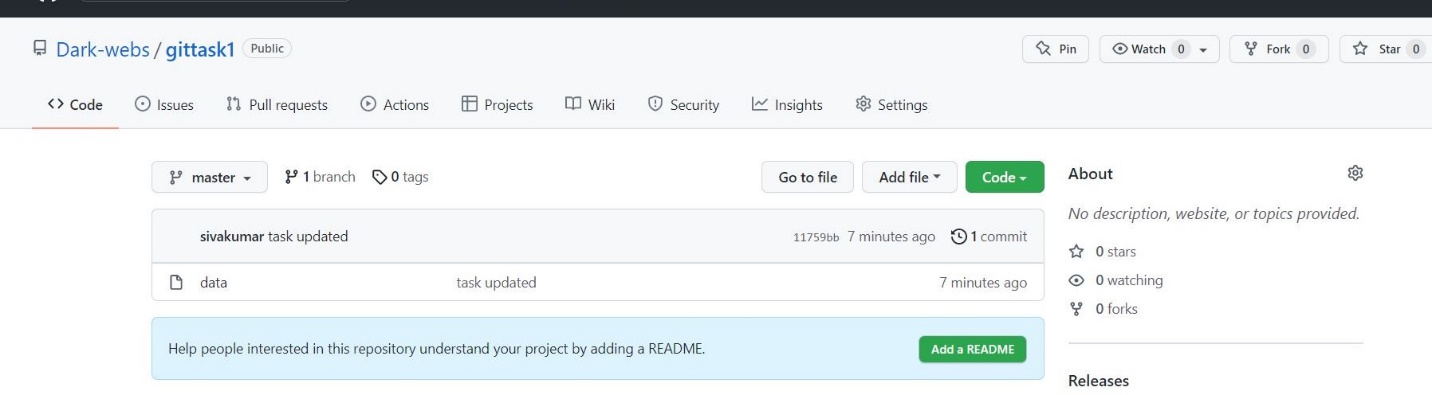
**[root@ip-172-31-85-184 task1]#**

**Now my file was pushed in the remote repository go to the github just refresh your repository page**



**After refresh it will display your data application this way we want do local to remote mapping**

**System.**

****