

Github

(How to add,commit | How to push from git to github | How to pull code from github)

Do SSH in your CMD

- Yum update -y (update your file)s
- Yum install git -y
- Which git (check git is installed or not)
- git -- version

```
ubuntu@ip-172-31-13-21:~$ git --version
git version 2.25.1
ubuntu@ip-172-31-13-21:~$ which git
/usr/bin/git
ubuntu@ip-172-31-13-21:~$ git status
fatal: not a git repository (or any of the parent directories): .git
```

- first create a directory ,directory name is satyam
- if I want to convert a directory in repository we use this command (git init)
- we use cat command to create a new file
e.x = cat >file1, after written some data in file, enter ctrl+d to save
- we use git status to see our file
- we use **git add** command to move our file in staging area.

```
ubuntu@ip-172-31-13-21:~$ sudo su
root@ip-172-31-13-21:/home/ubuntu# ls
root@ip-172-31-13-21:/home/ubuntu# mkdir satyam
root@ip-172-31-13-21:/home/ubuntu# ls
satyam
root@ip-172-31-13-21:/home/ubuntu# cd satyam
root@ip-172-31-13-21:/home/ubuntu/satyam# git init
Initialized empty Git repository in /home/ubuntu/satyam/.git/
root@ip-172-31-13-21:/home/ubuntu/satyam# ls
root@ip-172-31-13-21:/home/ubuntu/satyam# cat satyam1
cat: satyam1: No such file or directory
root@ip-172-31-13-21:/home/ubuntu/satyam# cat >file1
hello my name is satyam
what is your name?root@ip-172-31-13-21:/home/ubuntu/satyam# git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  file1

nothing added to commit but untracked files present (use "git add" to track)
root@ip-172-31-13-21:/home/ubuntu/satyam# git add
Nothing specified, nothing added.
Maybe you wanted to say 'git add .'
root@ip-172-31-13-21:/home/ubuntu/satyam# git add file1
root@ip-172-31-13-21:/home/ubuntu/satyam# git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
  new file:   file1
```

- We use **git commit -m** to move staging to production, see there are one file changed and two new file are insertion

```

root@ip-172-31-13-21:/home/ubuntu/satyam# git commit -m "hello india "
[master (root-commit) fb42f45] hello india
Committer: root <root@ip-172-31-13-21.ap-south-1.compute.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 2 insertions(+)
create mode 100644 file1
root@ip-172-31-13-21:/home/ubuntu/satyam#

```

- git status commands to know in which branch you are working
- we use git log to know commit id and commit details

```

root@ip-172-31-13-21:/home/ubuntu/satyam# git status
On branch master
nothing to commit, working tree clean
root@ip-172-31-13-21:/home/ubuntu/satyam# git log
commit fb42f4501a69638a1393d71fd9ce13b2df8452d0 (HEAD -> master)
Author: root <root@ip-172-31-13-21.ap-south-1.compute.internal>
Date: Thu Sep 23 10:32:06 2021 +0000

    hello india
root@ip-172-31-13-21:/home/ubuntu/satyam#

```

- now, we see our file is move on production server ,we see our commit id ,date and time and we also see a message hello india .

```

root@ip-172-31-13-21:/home/ubuntu/satyam# git status
On branch master
nothing to commit, working tree clean
root@ip-172-31-13-21:/home/ubuntu/satyam#

```

- now you see your staging area will be clean because you move your file staging to production.

```

root@ip-172-31-13-21:/home/ubuntu/satyam# git log
commit fb42f4501a69638a1393d71fd9ce13b2df8452d0 (HEAD -> master)
Author: root <root@ip-172-31-13-21.ap-south-1.compute.internal>
Date: Thu Sep 23 10:32:06 2021 +0000

    hello india
root@ip-172-31-13-21:/home/ubuntu/satyam# git status
On branch master
nothing to commit, working tree clean
root@ip-172-31-13-21:/home/ubuntu/satyam# git show fb42f45
commit fb42f4501a69638a1393d71fd9ce13b2df8452d0 (HEAD -> master)
Author: root <root@ip-172-31-13-21.ap-south-1.compute.internal>
Date: Thu Sep 23 10:32:06 2021 +0000

    hello india

diff --git a/file1 b/file1
new file mode 100644
index 0000000..a5b5be0
--- /dev/null
+++ b/file1
@@ -0,0 +1,2 @@
+hello my name is satyam
+what is your name?
\ No newline at end of file
root@ip-172-31-13-21:/home/ubuntu/satyam#

```

- if you want to see your file1 data you write a command
example => (git show **commit-id**)

git show fb42f4501a69638a1393d71fd9ce13b2df8452d0

```

root@ip-172-31-7-193:/home/ubuntu/satyam# git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Satyamtripathi1996/testing-repository.git
git push -u origin main
On branch master
nothing to commit, working tree clean
root@ip-172-31-7-193:/home/ubuntu/satyam# git branch -M main
root@ip-172-31-7-193:/home/ubuntu/satyam# git remote add origin https://github.com/Satyamtripathi1996/testing-repository.git
root@ip-172-31-7-193:/home/ubuntu/satyam# git push -u origin main
Username for 'https://github.com': satyam
Password for 'https://satyam@github.com':
remote: Support for password authentication was removed on August 13, 2021. Please use a personal access token instead.
remote: Please see https://github.blog/2020-12-15-token-authentication-requirements-for-git-operations/ for more information.
fatal: Authentication failed for 'https://github.com/Satyamtripathi1996/testing-repository.git/'

```

- If error come please go to this link = <https://www.theserverside.com/blog/Coffee-Talk-Java-News-Stories-and-Opinions/Fix-GitHubs-support-for-password-authentication-was-removed-error>

How to generate a GitHub personal access token

To generate a personal access token in GitHub and fix the 'support for password authentication removed' error, follow these steps:

1. Log into GitHub with your username and password
2. Navigate to your GitHub account settings
3. Scroll down and click 'Developer settings' in the list of links to the left
4. Click the Personal access tokens link
5. Click the 'Generate new token' button
6. Add a 'Note' to describe the tokens usage
7. Set an expiration date for the GitHub personal access token
8. Select the appropriate authentication scope
9. Click the 'Generate token' button

Then copy your personal access token before you navigate away from the page, as the token's alphanumeric value will not be displayed again. The next time you push to GitHub, provide the personal access token instead of the password, and you will have fixed the 'support for password authentication was removed' error.



```
root@ip-172-31-7-193:/home/ubuntu/satyam# git push -u origin main
Username for 'https://github.com': git push origin
Password for 'https://git push origin@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 246 bytes | 246.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/Satyamtripathi1996/testing-reposiory.git
* [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
root@ip-172-31-7-193:/home/ubuntu/satyam#
```

(root@ip-172-31-7-193:/home/Ubuntu/satyam# git push -u origin main)

→ It means you push your file to staging in main branch

```
root@ip-172-31-7-193:/home/ubuntu/satyam# git push -u origin main
Username for 'https://github.com': satyam
Password for 'https://satyam@github.com':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 279 bytes | 279.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/Satyamtripathi1996/testing-reposiory.git
d499c3d..71fa182 main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
root@ip-172-31-7-193:/home/ubuntu/satyam#
```

→ If you change something in your local repository file and you want to commit your changes in your main github repository file, firstly run this command

→ root@ip-172-31-7-193:/home/ubuntu/satyam# **git add file1**

→ root@ip-172-31-7-193:/home/ubuntu/satyam# **git commit -m file1**

→ root@ip-172-31-7-193:/home/Ubuntu/satyam# git push -u origin main

```
Username for 'https://github.com': satyam
Password for 'https://satyam@github.com':
```

→ If you don't know password again enter your access token in password

→ Now, go and check your main repository has been updated.

DONE SUCCESSFULLY

| files = You can configure Git to ignore files you don't want to check in to GitHub.

an create a *.gitignore* file in your repository's root directory to tell Git which files and directories to ignore n you make a commit. To share the ignore rules with other users who clone the repository, commit *gitignore* file in to your repository.

```
ubuntu@ip-172-31-38-174:~$ sudo su
root@ip-172-31-38-174:/home/ubuntu# ls
satyam
root@ip-172-31-38-174:/home/ubuntu# cd satyam/
root@ip-172-31-38-174:/home/ubuntu/satyam# ls
file1
root@ip-172-31-38-174:/home/ubuntu/satyam# touch file2.txt
root@ip-172-31-38-174:/home/ubuntu/satyam# vi touch.txt
root@ip-172-31-38-174:/home/ubuntu/satyam# root@ip-172-31-38-174:~$
>
root@ip-172-31-38-174:/home/ubuntu/satyam# ls
file1 file2.txt touch.txt
root@ip-172-31-38-174:/home/ubuntu/satyam# vi .gitignore
```

```
root@ip-172-31-38-174:/home/ubuntu/satyam
*.txt
```

➔ Suppose we don't want to commit any .txt or any type of file extension so we have option to write file name or file extension in .gitignore file and save it they will ignore your file when you commit it

```
root@ip-172-31-38-174:/home/ubuntu/satyam
root@ip-172-31-38-174:/home/ubuntu/satyam# vi .gitignore
root@ip-172-31-38-174:/home/ubuntu/satyam# git add .gitignore
root@ip-172-31-38-174:/home/ubuntu/satyam# git commit -m "ignore file format"
[master f2962e3] ignore file format
Committer: root <root@ip-172-31-38-174.ap-south-1.compute.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 1 insertion(+)
create mode 100644 .gitignore
root@ip-172-31-38-174:/home/ubuntu/satyam# git status
On branch master
nothing to commit, working tree clean
root@ip-172-31-38-174:/home/ubuntu/satyam# git log
commit f2962e3e008e02dc20c8e27320c94fe38f89b080 (HEAD -> master)
Author: root <root@ip-172-31-38-174.ap-south-1.compute.internal>
Date:   Wed Oct 6 06:39:08 2021 +0000

    ignore file format

commit d17dd015d37376a77f539257f188ce68acc76638 (origin/master)
Author: root <root@ip-172-31-38-174.ap-south-1.compute.internal>
Date:   Fri Oct 1 10:12:05 2021 +0000

    file1
root@ip-172-31-38-174:/home/ubuntu/satyam#
```



```
root@ip-172-31-38-174:/home/ubuntu/satyam# touch file1.txt file2.txt file3.java file4.css
root@ip-172-31-38-174:/home/ubuntu/satyam# ls
file1 file2.txt file3.java file4.css touch.txt
root@ip-172-31-38-174:/home/ubuntu/satyam# git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file3.java
        file4.css

nothing added to commit but untracked files present (use "git add" to track)
root@ip-172-31-38-174:/home/ubuntu/satyam# git add.
git: 'add.' is not a git command. See 'git --help'.

The most similar command is
  add
root@ip-172-31-38-174:/home/ubuntu/satyam# git add.
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   file3.java
        new file:   file4.css

root@ip-172-31-38-174:/home/ubuntu/satyam# git commit -m "ignore .txt file"
[master cc7ca18] ignore .txt file
Committer: root <root@ip-172-31-38-174.ap-south-1.compute.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:
    git config --global --edit

After doing this, you may fix the identity used for this commit with:
    git commit --amend --reset-author

2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file3.java
create mode 100644 file4.css
root@ip-172-31-38-174:/home/ubuntu/satyam# git log
commit cc7ca18beb9c59610d347f3aa7781d02568ac32 (HEAD -> master)
Author: root <root@ip-172-31-38-174.ap-south-1.compute.internal>
Date:   Wed Oct 6 07:47:30 2021 +0000

    ignore .txt file

commit f2962e3c008e02dc20c8c27320c94fe38f98b080
Author: root <root@ip-172-31-38-174.ap-south-1.compute.internal>
Date:   Wed Oct 6 06:39:08 2021 +0000

    ignore file format

commit d17dd915d9727677753925f188ce8acc70638 (origin/master)
Author: root <root@ip-172-31-38-174.ap-south-1.compute.internal>
Date:   Fri Oct 1 10:12:05 2021 +0000

    file1

root@ip-172-31-38-174:/home/ubuntu/satyam# git show cc7ca18beb9c5
commit cc7ca18beb9c59610d347f3aa7781d02568ac32 (HEAD -> master)
Author: root <root@ip-172-31-38-174.ap-south-1.compute.internal>
Date:   Wed Oct 6 07:47:30 2021 +0000

    ignore .txt file

diff --git a/file3.java b/file3.java
new file mode 100644
index 0000000..e99de29
diff --git a/file4.css b/file4.css
new file mode 100644
index 0000000..e99de29
root@ip-172-31-38-174:/home/ubuntu/satyam#
```

```
root@ip-172-31-38-174:/home/ubuntu/satyam# git log --oneline
cc7ca18 (HEAD -> master) ignore .txt file
f2962e3 ignore file format
d17dd91 (origin/master) file1
root@ip-172-31-38-174:/home/ubuntu/satyam#
```

➔ If you want to see all commit in one line you use this command :- **root@ip-172-31-38-174:/home/ubuntu/satyam# git log --oneline**

create a username & email

- ➔ git config --global user.name "satyam"
- ➔ git config --global user.email satyamchamptripathi@gmail.com
- ➔ check username & email
- ➔ git config --list

```
root@ip-172-31-7-193:/home/ubuntu/satyam# git config --list
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
remote.origin.url=https://github.com/Satyamtripathi1996/testing-reposiory.git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
branch.main.remote=origin
branch.main.merge=refs/heads/main
root@ip-172-31-7-193:/home/ubuntu/satyam#
```

if you want to know in which branch you are working right now

- ➔ git branch
 - * master (if * will come in your branch name it means you are inside this branch)
 - Satyam
 - India

If you want to create a new branch use this command

- ➔ git branch Allahabad

if you want to change your branch :-

- ➔ Git checkout Allahabad

Master

- * Allahabad (Now you are inside a Allahabad branch)