



```
ubuntu@ip-172-31-38-174:~$ sudo su
root@ip-172-31-38-174:/home/ubuntu# ls
satyam
```

→ Workspace name = satyam

root@ip-172-31-38-174:/home/ubuntu/satyam# git branch (check branch detail)

root@ip-172-31-38-174:/home/ubuntu/satyam# git branch satyam1 (create new branch = satyam1)

```
root@ip-172-31-4-43:/home/ubuntu/satyam# git branch
* main
root@ip-172-31-4-43:/home/ubuntu/satyam# git branch satyam1
```

```
root@ip-172-31-4-43:/home/ubuntu/satyam# git branch
* main
  satyam1
```

```
root@ip-172-31-4-43:/home/ubuntu/satyam# touch india1.txt
root@ip-172-31-4-43:/home/ubuntu/satyam# vi india1.txt
root@ip-172-31-4-43:/home/ubuntu/satyam# git add .
root@ip-172-31-4-43:/home/ubuntu/satyam# git commit -m
error: switch `m` requires a value
root@ip-172-31-4-43:/home/ubuntu/satyam# git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   india1.txt
```

```
root@ip-172-31-4-43:/home/ubuntu/satyam
hello india ,
```

→ root@ip-172-31-38-174:/home/ubuntu/satyam# git checkout branch1 (change your workspace master branch to satyam1)

```
root@ip-172-31-4-43:/home/ubuntu/satyam# git checkout satyam1
Switched to branch 'satyam1'
root@ip-172-31-4-43:/home/ubuntu/satyam# touch india1.txt
root@ip-172-31-4-43:/home/ubuntu/satyam# vi india1.txt
root@ip-172-31-4-43:/home/ubuntu/satyam# git status
On branch satyam1
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        india1.txt

nothing added to commit but untracked files present (use "git add" to track)
root@ip-172-31-4-43:/home/ubuntu/satyam# git add.
git: 'add.' is not a git command. See 'git --help'.

The most similar command is
  add
root@ip-172-31-4-43:/home/ubuntu/satyam# git add .
root@ip-172-31-4-43:/home/ubuntu/satyam# git status
On branch satyam1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   india1.txt
```

```
root@ip-172-31-4-43: /home/ubuntu/satyam
```

```
hello allahabad
```

```
root@ip-172-31-4-43:/home/ubuntu/satyam# git commit -m "testing conflict 2"
[satyam1 a3500d6] testing conflict 2
1 file changed, 1 insertion(+)
 create mode 100644 india1.txt
root@ip-172-31-4-43:/home/ubuntu/satyam# git status
On branch satyam1
nothing to commit, working tree clean
```

## NOTES START

---

→ If you want to see all the files in branch you use this command

`git ls-files`

```
root@ip-172-31-4-43:/home/ubuntu/satyam# git ls-files
bharat.txt
bharat1.txt
demofile
india1.txt
root@ip-172-31-4-43:/home/ubuntu/satyam#
```

→ If you want to delete files inside the branch

`root@ip-172-31-4-43:/home/ubuntu/satyam# git rm -r file name`

```
root@ip-172-31-4-43:/home/ubuntu/satyam# git branch
* main
  satyam1
root@ip-172-31-4-43:/home/ubuntu/satyam# git ls-files
bharat.txt
bharat1.txt
demofile
india1.txt
root@ip-172-31-4-43:/home/ubuntu/satyam# git rm -r bharat.txt
rm 'bharat.txt'
root@ip-172-31-4-43:/home/ubuntu/satyam# git ls-files
bharat1.txt
demofile
india1.txt
root@ip-172-31-4-43:/home/ubuntu/satyam# git rm -r bharat1.txt
rm 'bharat1.txt'
root@ip-172-31-4-43:/home/ubuntu/satyam# git ls-files
demofile
india1.txt
root@ip-172-31-4-43:/home/ubuntu/satyam#
```

## NOTES END

---

# Merge

- You Can't merge branches of different Repositories.
- We use pulling Mechanism to merge branches

**git merge <branch name>**

```
root@ip-172-31-4-43:/home/ubuntu/satyam# git merge satyam1
CONFLICT (add/add): Merge conflict in india1.txt
Auto-merging india1.txt
Automatic merge failed; fix conflicts and then commit the result.
root@ip-172-31-4-43:/home/ubuntu/satyam# vi india1.txt
```

## Git conflict

- When same name file having different content in different branches if you do merge, conflicts occurs come (Resolve conflict then add and commit).
- Change something in file and then again save it

```
root@ip-172-31-4-43:/home/ubuntu/satyam
hello india ,
Hello Allahabad
```

- Save file
- Again run add command because you edit a file  
**git add .**
- Again run commit command because you edit a file  
**git commit -m "resolved conflict"**
- Now, again run merge command  
**git merge satyam1**
- Conflict issue resolved

- To verify the merge status

**git log**

```

root@ip-172-31-4-43: /home/ubuntu/satyam
root@ip-172-31-4-43:/home/ubuntu/satyam# git log
commit 2dd17ea146b8e3c8b98e64c9346c241b77102637 (HEAD -> main)
Merge: ecef6c2 a3500d6
Author: satyam <Satyamchamptripathi@gmail.com>
Date: Tue Oct 12 12:47:42 2021 +0000

    conflict issue resolved

commit a3500d6f5e459c8d641632dbe96afd019715ab70 (satyam1)
Author: satyam <Satyamchamptripathi@gmail.com>
Date: Tue Oct 12 12:41:28 2021 +0000

    testing conflict 2

commit ecef6c239f159d4f4c6a4232a95cf1f04c00512e
Author: satyam <Satyamchamptripathi@gmail.com>
Date: Tue Oct 12 12:30:27 2021 +0000
:...skipping...
commit 2dd17ea146b8e3c8b98e64c9346c241b77102637 (HEAD -> main)
Merge: ecef6c2 a3500d6
Author: satyam <Satyamchamptripathi@gmail.com>
Date: Tue Oct 12 12:47:42 2021 +0000

    conflict issue resolved

commit a3500d6f5e459c8d641632dbe96afd019715ab70 (satyam1)
Author: satyam <Satyamchamptripathi@gmail.com>
Date: Tue Oct 12 12:41:28 2021 +0000

    testing conflict 2

commit ecef6c239f159d4f4c6a4232a95cf1f04c00512e
Author: satyam <Satyamchamptripathi@gmail.com>
Date: Tue Oct 12 12:30:27 2021 +0000

    testing conflict

```

➔ To view all log in one line

**git log --oneline**

```

root@ip-172-31-4-43:/home/ubuntu/satyam# git log --oneline
2dd17ea (HEAD -> main) conflict issue resolved
a3500d6 (satyam1) testing conflict 2
ecef6c2 testing conflict

```

# Git Stashing

- ➔ Stashing is like a temporary storage.
- ➔ If you are doing some work inside a branch and someone tells you to this new work first, then do your old work, so we do have to create a new branch to do new work, you put your old code in stash ,because stash is a temporary storage where we save your files.
- ➔ Suppose you are implementing a new feature for your product, your code is in progress and suddenly a customer escalation comes because of this, you have to keep aside your new feature work for few hours.
- ➔ You cannot commit your partial code and also cannot throw away your changes so you need some temporary storage, when you can store your partial changes and later on commit .
- ➔ To stash an item (only applies to modified files not new files) .
- ➔ To stash an item  
`git stash`
- ➔ To see stashed items list  
`git stash list`
- ➔ To apply stashed items  
`git stash apply stash@{0}` (whatever stash you want put serial number)  
0 serial number is the new stash file
- ➔ To clear the stash items  
`git stash clear`





```
root@ip-172-31-43-31:/home/ubuntu/satyam# git stash list
stash@{0}: WIP on master: 6712b2a this is a demo file
root@ip-172-31-43-31:/home/ubuntu/satyam# vi tripathi1.txt
root@ip-172-31-43-31:/home/ubuntu/satyam#
```

```
root@ip-172-31-43-31:/home/ubuntu/satyam
```

bhai meri jo pehale tripathi1.txt vali file thii usme mene jo likha thaa vo stash storage mein chala gaya hai, jaise hi mene stash vali command chalayi vo mere stash kein local storage mein jaake save ho gayi hai

-> ab yein vali file me agar mein kuch naya lihhana chah raha toh likh sakta hu  
-> aapko yein sochane ki jarurat nahi padn chahihe ki agar aapne apni file stash mein daal di toh yaha tripathi1.txt name sein file show kaise ho rahi hai aapke new workspace mein toh by default shayad purani file ban jati hai bas uska data kuch nahi rehata new file mein..

```
root@ip-172-31-43-31:/home/ubuntu/satyam# git stash apply stash@{0}
error: Your local changes to the following files would be overwritten by merge:
    tripathi1.txt
Please commit your changes or stash them before you merge.
Aborting
root@ip-172-31-43-31:/home/ubuntu/satyam# git add .
root@ip-172-31-43-31:/home/ubuntu/satyam# git commit-m " this is my stash file data"
git: 'commit-m' is not a git command. See 'git --help'.

The most similar command is
    commit-tree
root@ip-172-31-43-31:/home/ubuntu/satyam# git stash apply stash@{0}
Auto-merging tripathi1.txt
CONFLICT (content): Merge conflict in tripathi1.txt
root@ip-172-31-43-31:/home/ubuntu/satyam# vi tripathi1.txt
root@ip-172-31-43-31:/home/ubuntu/satyam# git stash apply stash@{0}
tripathi1.txt: needs merge
root@ip-172-31-43-31:/home/ubuntu/satyam#
root@ip-172-31-43-31:/home/ubuntu/satyam# vi tripathi1.txt
root@ip-172-31-43-31:/home/ubuntu/satyam# root@ip-172-31-43-31:/home/ubuntu/satyam#
root@ip-172-31-43-31:/home/ubuntu/satyam#
```

➔ git stash clear

```
root@ip-172-31-43-31:/home/ubuntu/satyam
```

```
root@ip-172-31-43-31:/home/ubuntu# cd satyam/
root@ip-172-31-43-31:/home/ubuntu/satyam# git stash list
stash@{0}: WIP on master: 6712b2a this is a demo file
root@ip-172-31-43-31:/home/ubuntu/satyam# git stash clear
root@ip-172-31-43-31:/home/ubuntu/satyam# git stash list
root@ip-172-31-43-31:/home/ubuntu/satyam#
```

If we want to clear our stash we use this command

# Git Reset

The term reset stands for undoing changes. The git reset command is used to reset the changes. The git reset command has three core forms of invocation. These forms are as follows.

- **Soft**
- **Mixed**
- **Hard**

Learn more : - <https://www.javatpoint.com/git-reset>

➔ git reset is a powerful command that is used to undo local changes to the state of a git repo .

To reset staging area :-

Single-file reset

```
git reset <filename>
```

whole staging reset:-

```
git reset.
```

➔ To reset the changes from both staging area and working directory data at a same time use this command

To reset the staging & working area both :-

```
Git reset --hard
```