



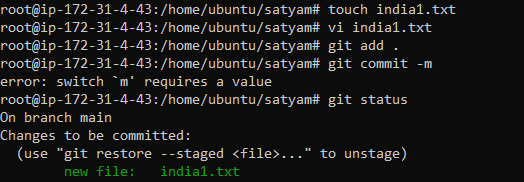
* Workspace name = satyam

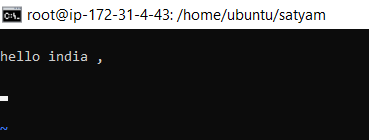
**root@ip-172-31-38-174:/home/ubuntu/satyam# git branch (check branch detail)**

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

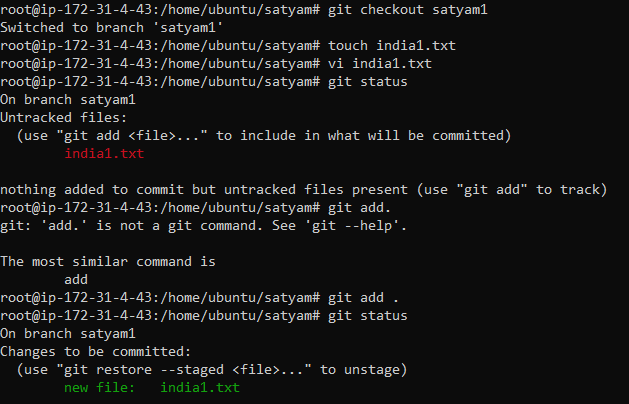
****

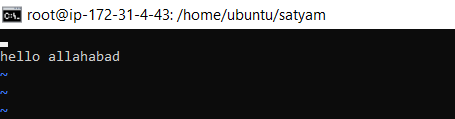
****

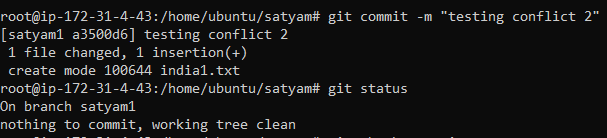
****

****

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

****

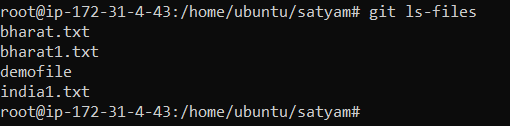
****

****

**NOTES START**

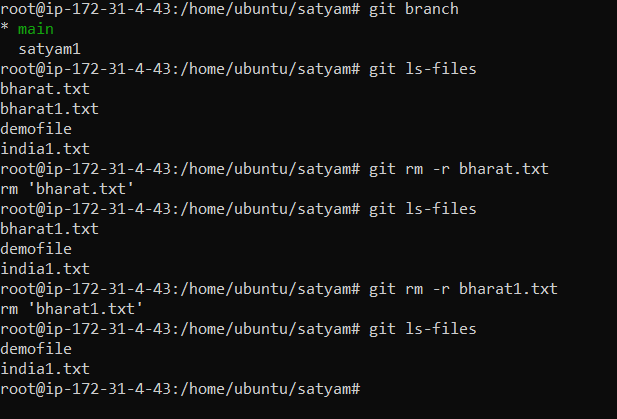
* If you want to see all the files in branch you use this command

git ls-files



* If you want to delete files inside the branch

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



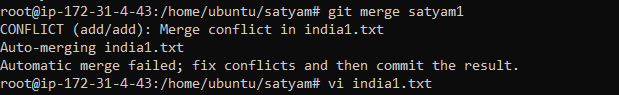
**NOTES END**

**---------------------------------------------------------------------------------------------------------------------------**

**Merge**

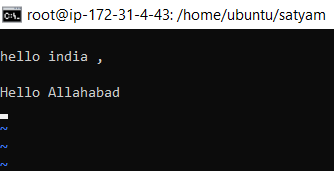
* You Can’t merge branches of different Repositories.
* We use pulling Mechanism to merge branches

**git merge <branch name>**

****

**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**

****

* **Save file**
* **Again run add command because you edit a file**

**git add .**

* **Again run commit command because you edit a file**

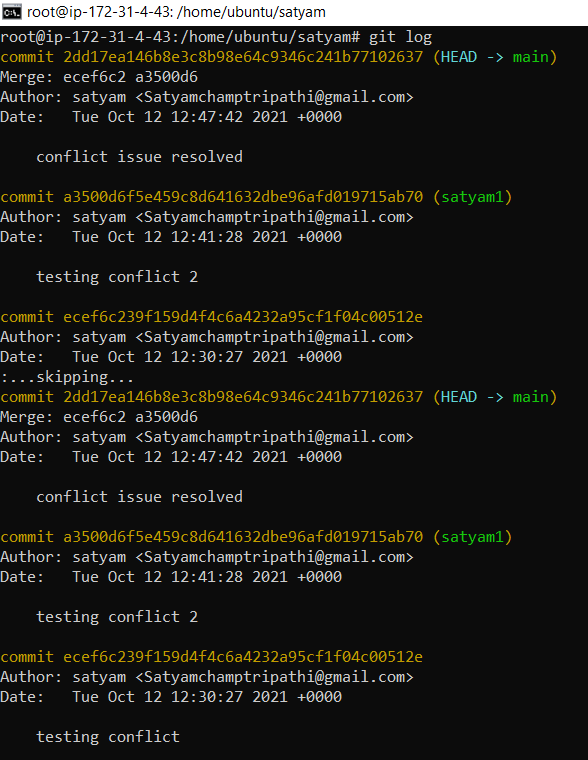
**git commit –m “resolved conflict”**

* **Now, again run merge command**

**git merge satyam1**

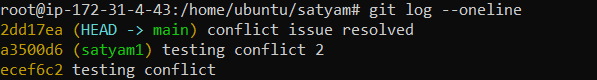
* **Conflict issue resolved**
* To verify the merge status

**git log**

****

* **To view all log in one line**

**git log --oneline**

****

**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

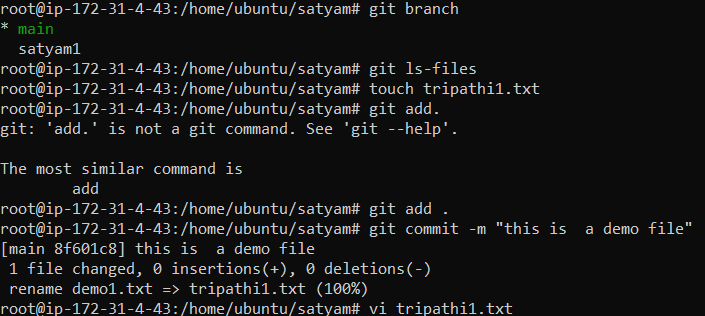
* To apply stashed items

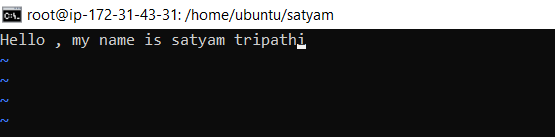
git stash apply stash@{0} (whatever stash you want put serial number)

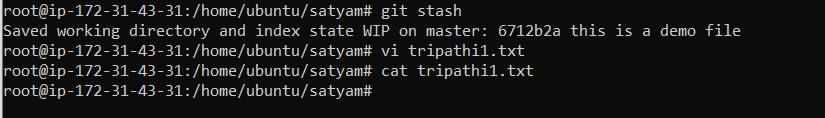
0 serial number is the new stash file

* To clear the stash items

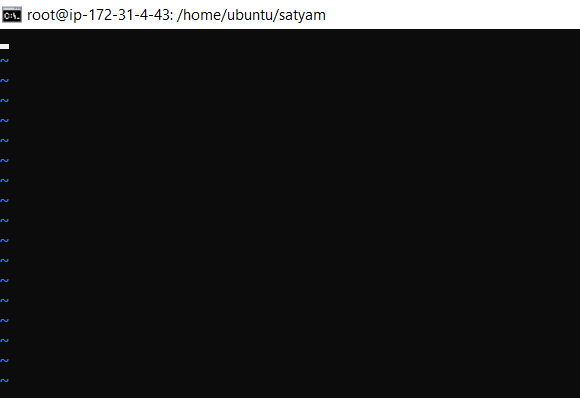
git stash clear

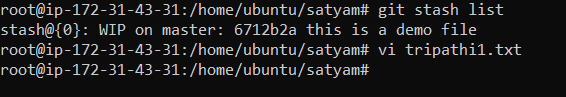


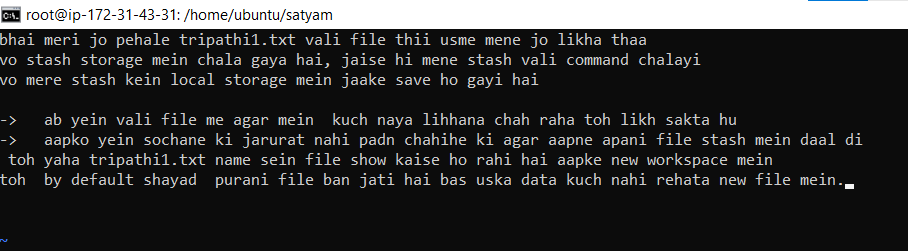


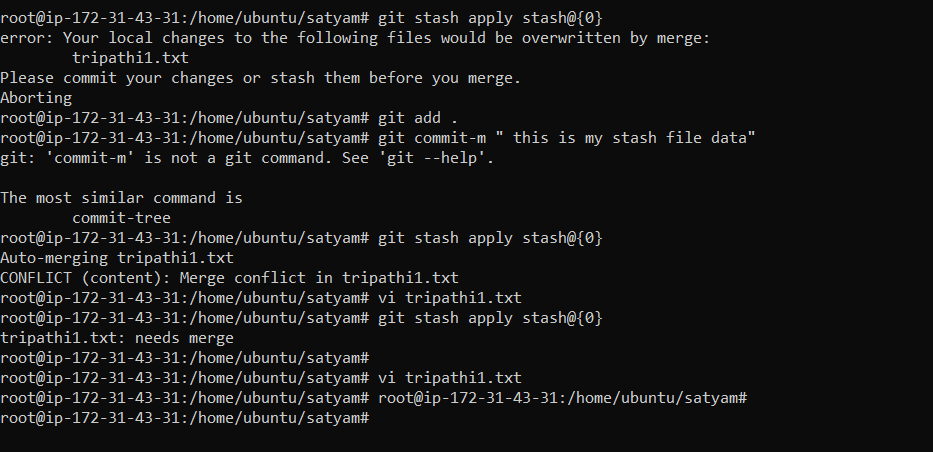


* Now, when we reopen our tripathi1.txt it is empty because our earlier work has been moved to stash temporary storage.

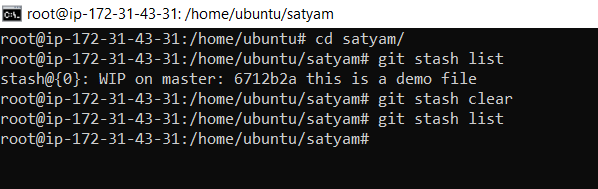








* **git stash clear**



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**](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 an working directory data at a same time usr this command

To reset the staging & working area both :-

Git reset –hard