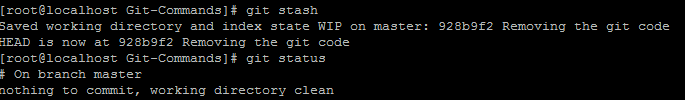
**Stash**

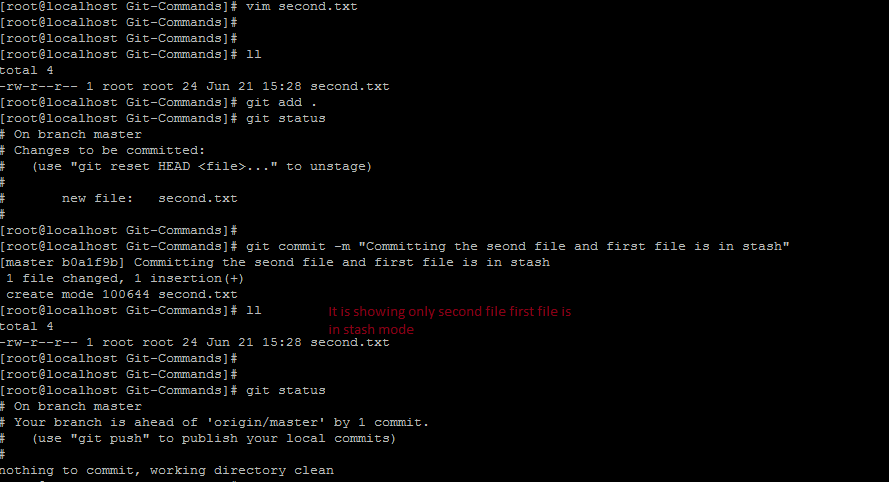
**git stash** temporarily save your changes that you have made to your working copy so you can work on something else, and then come back and re-apply them later on. Stashing is handy if you need to quickly switch context and work on something else, but you're mid-way through a code change and aren't quite ready to commit.

**Important:-** we can use git stash when the files are added in track mode(staging area).

When we apply git stash it save the file in work in progress mode inside git.



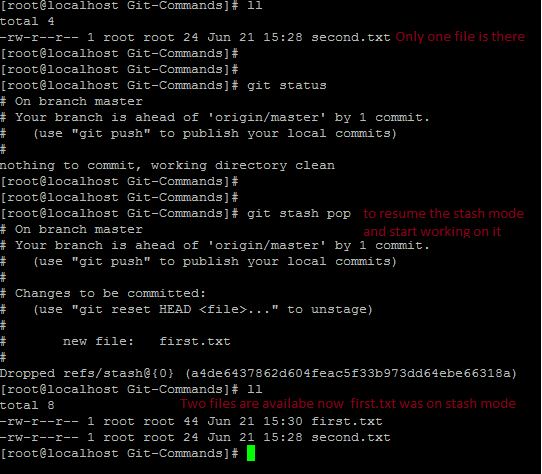
After applying git stash working directory become clean and you can start working on something else. In our case after applying git stash we have created the second file. First file stash area so it is not displaying.



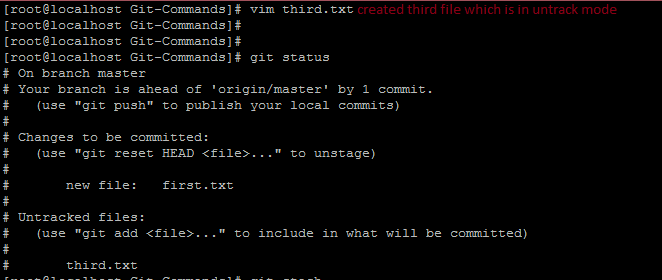
To resume the stash we can use command as :-

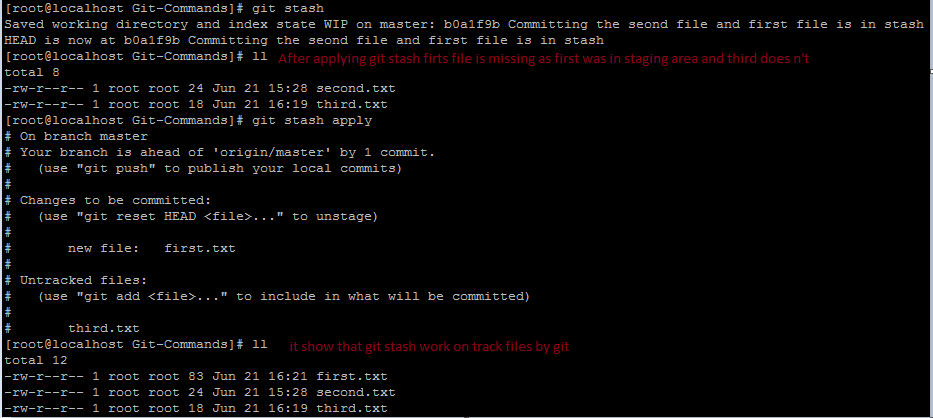
**git stash pop or git stash apply**

I have used git stash pop to resume git stash



Here in this on which we have added on file third.txt which was on untrack mode. So after applying git stash no effect was observed on it. Only the track file which was not committed got disappare.



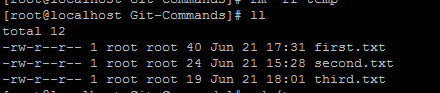


**clean**

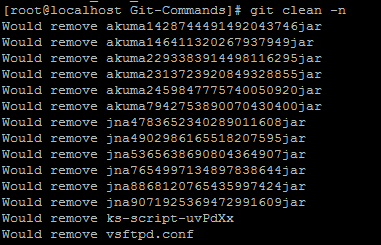
**git clean** is used to remove the untack file and folder from working directory.it is used when you have multiple files and folders in working directory and in untrack mode.



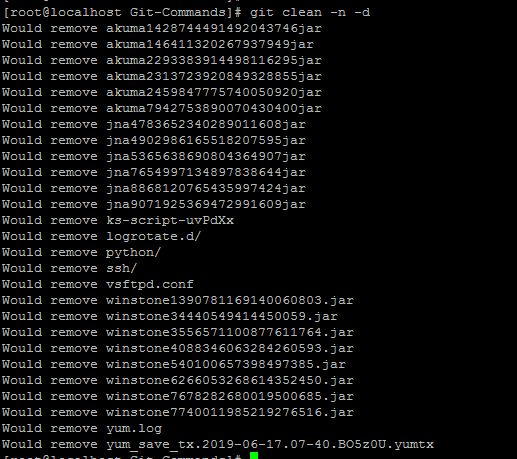
We cannot run git clean it will ask for flag option by displaying fatal error message.



git clean –n // list the file which would be remove

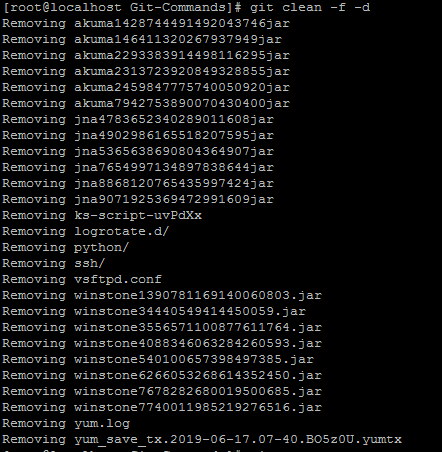


git clean –n –d // list the file and folders which would remove



git clean –f // it will remove the files

git clean –f –d // it will remove both files and directory.



Now after removing untrack files and folder.

