States in GIT:-

Working Directory:

This is where you make modifications to your files.
Files in this state are considered to be in the "untracked" or "modified" state.

Staging Area (Index):

After making changes in the working directory, you use the git add command to stage these changes.
Staging allows you to group related changes together before committing them.
Files in this state are considered to be in the "staged" state.

Local Repository:

After staging your changes, you use the git commit command to save them permanently to the Git repository.
Files in this state are considered to be in the "committed" state.

levels of configuration in GIT:-

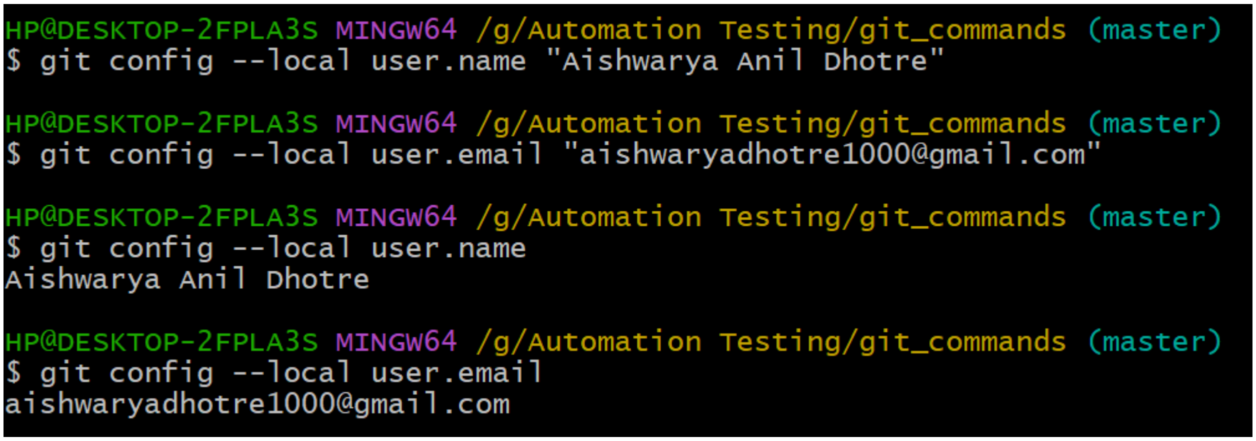
- **Repository (Local):-**
  - Stored in "/g/Automation Testing/git_commands/.git/config"
  - These settings are specific to a single Git repository.
  - They are stored in the .git/config file within the repository.
  - Local configurations override global configurations for that specific repository.
  - You can view and edit local-level configurations directly by opening the .git/config file or by using the git config command without any flag, which automatically refers to the local configuration for the current repository.

- **User Account (Global):-**
  - Stored in - "~/.gitconfig"
  - These settings are specific to a particular user on the system and apply to all Git repositories for that user.
  - They are stored in the user's home directory, usually in the .gitconfig file.

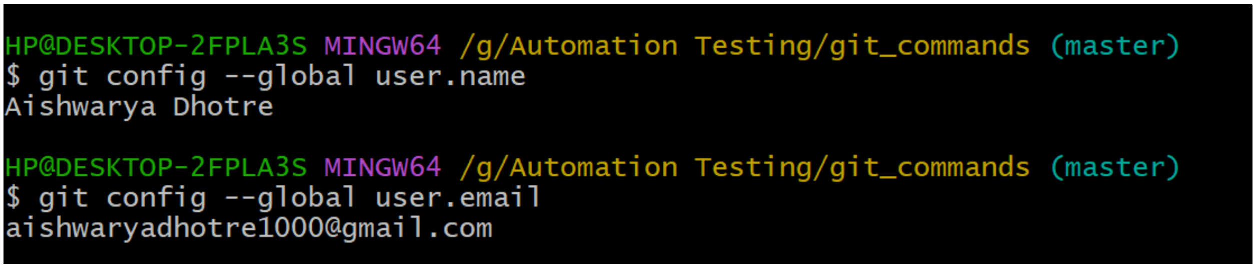
  - You can view and edit global-level configurations using the git config command with the --global flag.
    `git config --global --edit`

- **System Level (Git Installation) :-**
  - Stored in - "/etc/gitconfig"
  - These settings apply to the entire Git system and are shared among all users on the machine.
  - They are stored in the gitconfig file within the Git installation directory.
  - You can view and edit system-level configurations using the git config command with the --system flag.
    `git config --system --edit`

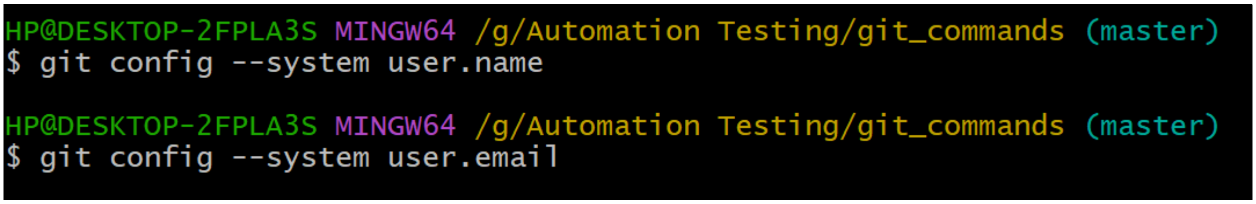
Config user name and password



```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git config --local user.name "Aishwarya Anil Dhotre"

HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git config --local user.email "aishwaryadhotre1000@gmail.com"

HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git config --local user.name
Aishwarya Anil Dhotre

HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git config --local user.email
aishwaryadhotre1000@gmail.com
```

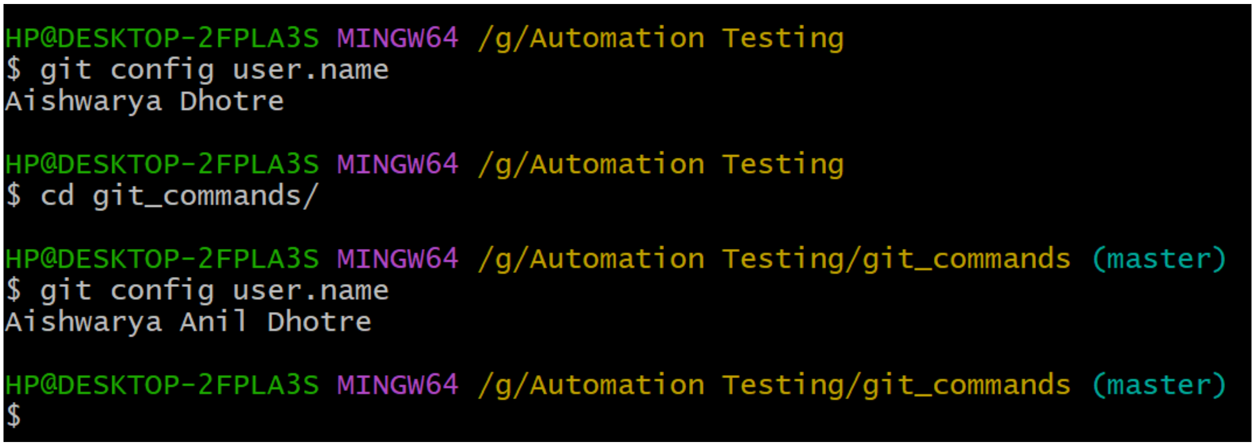


```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git config --global user.name
Aishwarya Dhotre

HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git config --global user.email
aishwaryadhotre1000@gmail.com
```



```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git config --system user.name

HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git config --system user.email
```

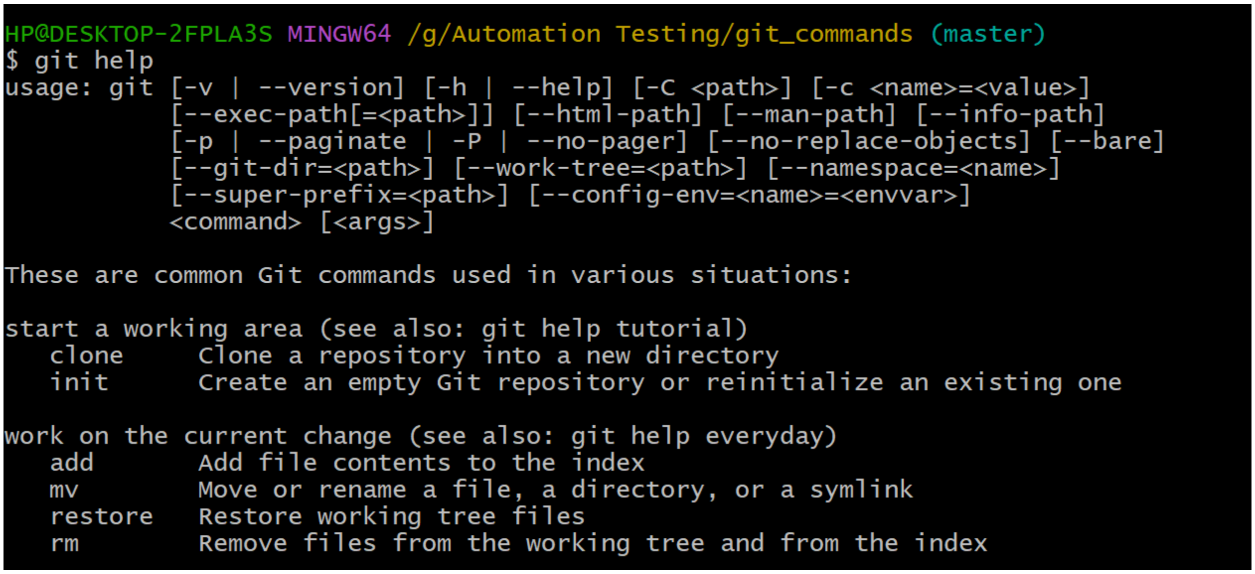
- The priority of configuration levels is such that local configurations take precedence over global configurations, which, in turn, take precedence over system configurations



```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing
$ git config user.name
Aishwarya Dhotre

HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing
$ cd git_commands/

HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git config user.name
Aishwarya Anil Dhotre

HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$
```

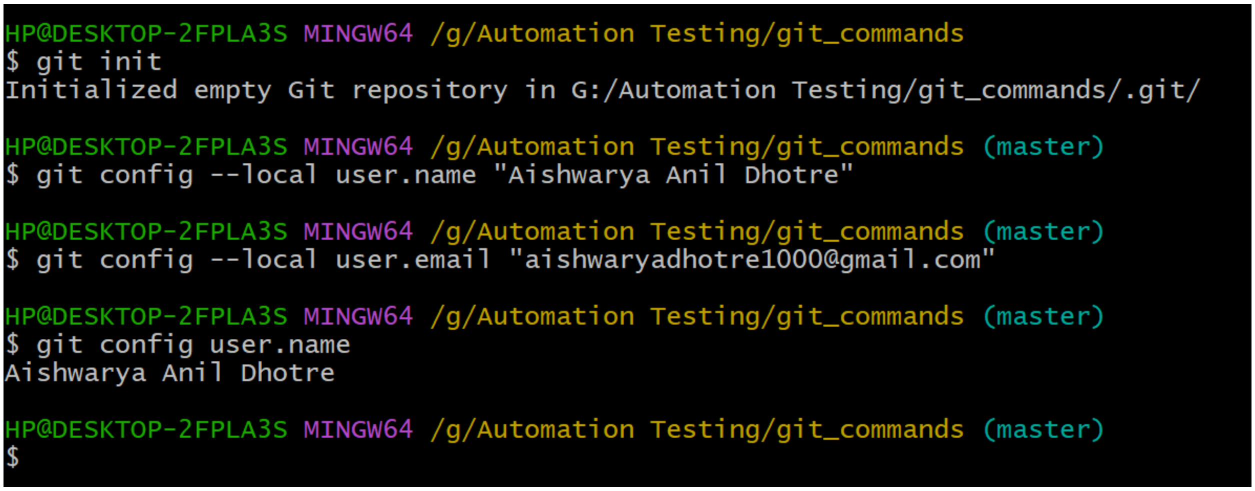
Git Commands

- Git help



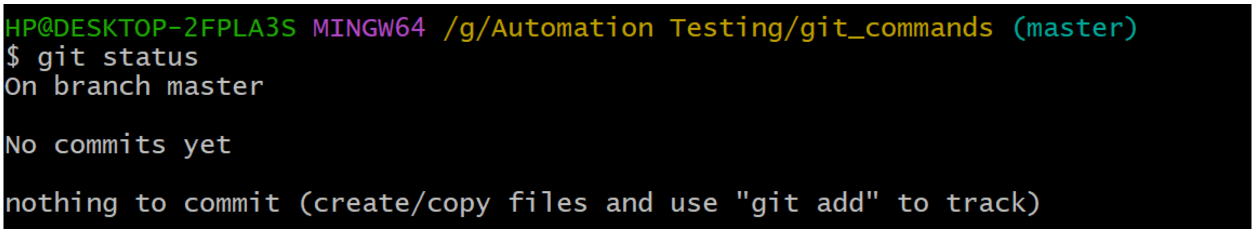
```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git help
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           [--super-prefix=<path>] [--config-env=<name>=<envvar>]
           <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
   clone     Clone a repository into a new directory
   init      Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
   add       Add file contents to the index
   mv        Move or rename a file, a directory, or a symlink
   restore   Restore working tree files
   rm        Remove files from the working tree and from the index
```

- Git init



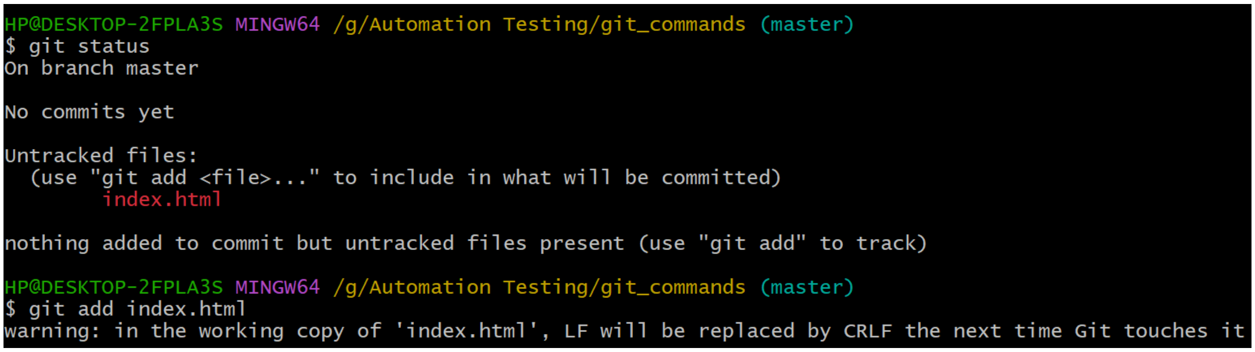
```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands
$ git init
Initialized empty Git repository in G:/Automation Testing/git_commands/.git/

HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git config --local user.name "Aishwarya Anil Dhotre"

HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git config --local user.email "aishwaryadhotre1000@gmail.com"

HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git config user.name
Aishwarya Anil Dhotre

HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$
```

- Git status

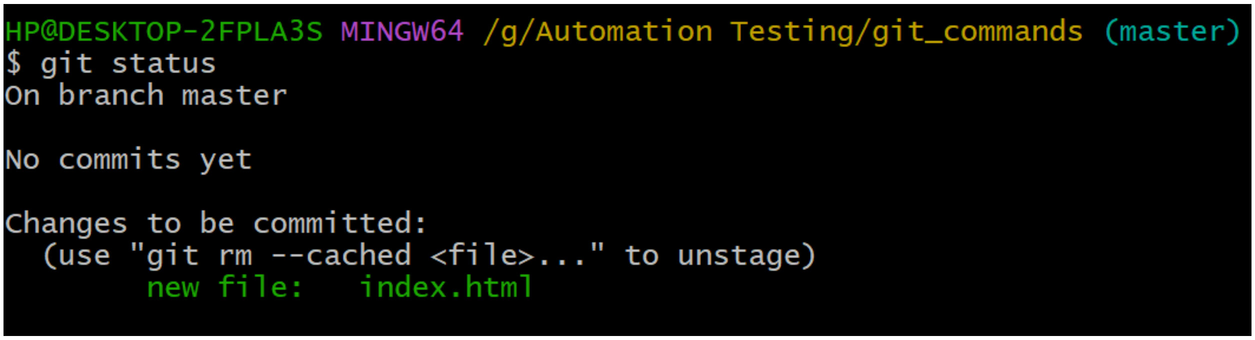


```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
```

- Git add

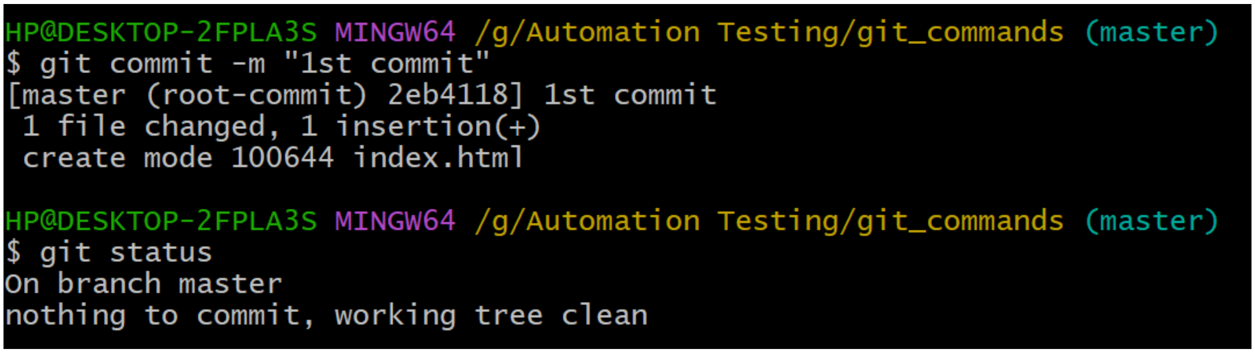


```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git status
On branch master

No commits yet

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

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

HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git add index.html
warning: in the working copy of 'index.html', LF will be replaced by CRLF the next time Git touches it
```

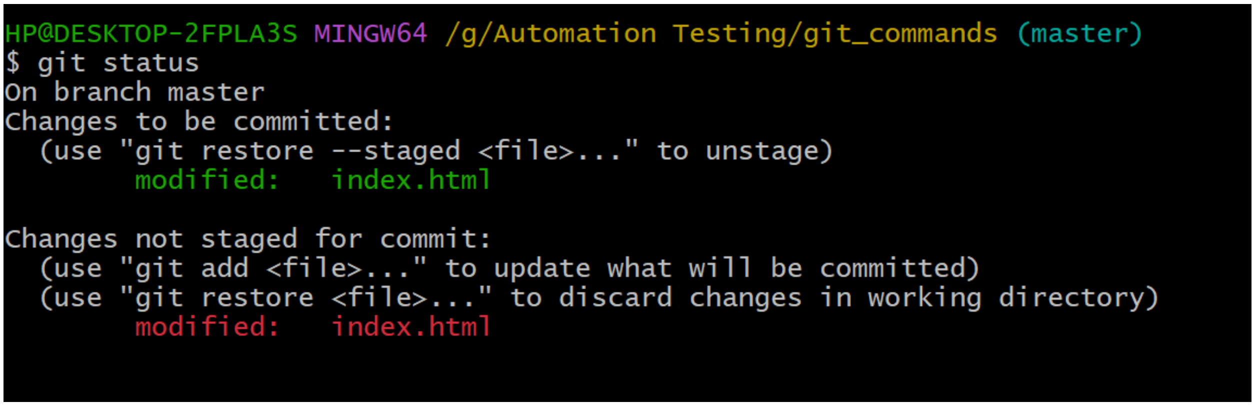


```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git status
On branch master

No commits yet

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

- Git commit

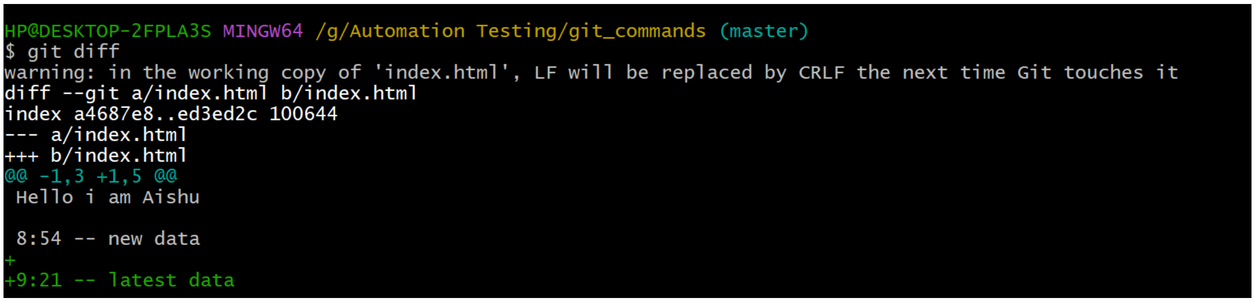


```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git commit -m "1st commit"
[master (root-commit) 2eb4118] 1st commit
 1 file changed, 1 insertion(+)
 create mode 100644 index.html

HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git status
On branch master
nothing to commit, working tree clean
```



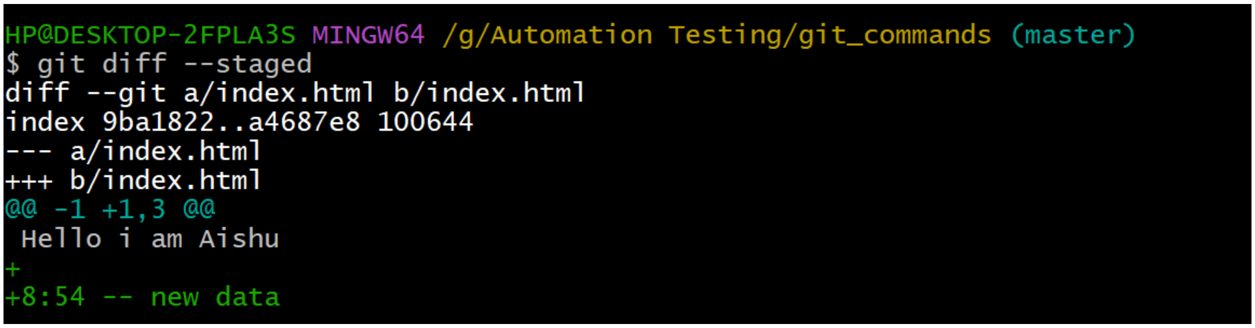
```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   index.html

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html
```

- Git diff: - compare the changes in the working area & staged area



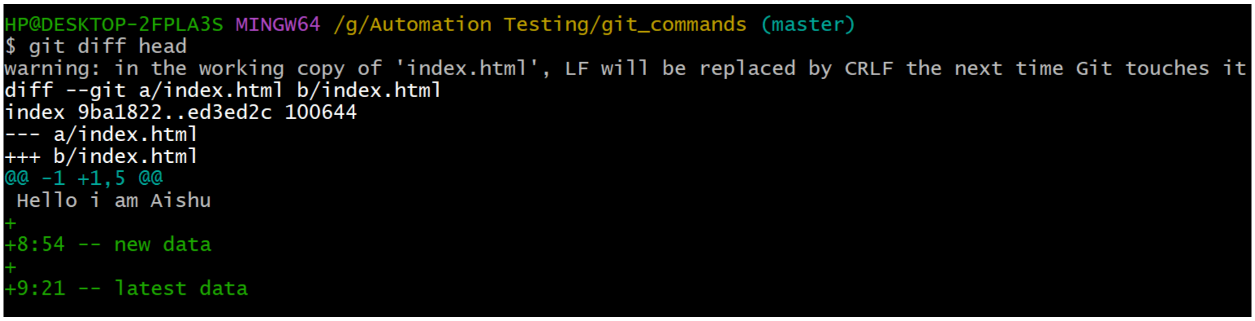
```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git diff
warning: in the working copy of 'index.html', LF will be replaced by CRLF the next time Git touches it
diff --git a/index.html b/index.html
index a4687e8..ed3ed2c 100644
--- a/index.html
+++ b/index.html
@@ -1,3 +1,5 @@
 Hello i am Aishu

 8:54 -- new data
+
+9:21 -- latest data
```

- Git diff –staged: - compare the changes in the staged area and repository area



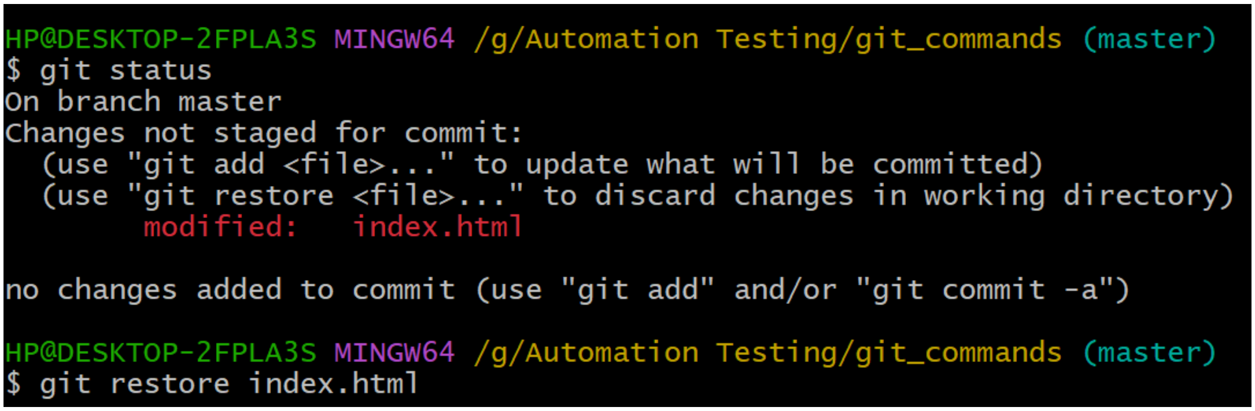
```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git diff --staged
diff --git a/index.html b/index.html
index 9ba1822..a4687e8 100644
--- a/index.html
+++ b/index.html
@@ -1 +1,3 @@
 Hello i am Aishu
+
+8:54 -- new data
```

- Git diff head:-   compare the changes in the working area and repository area



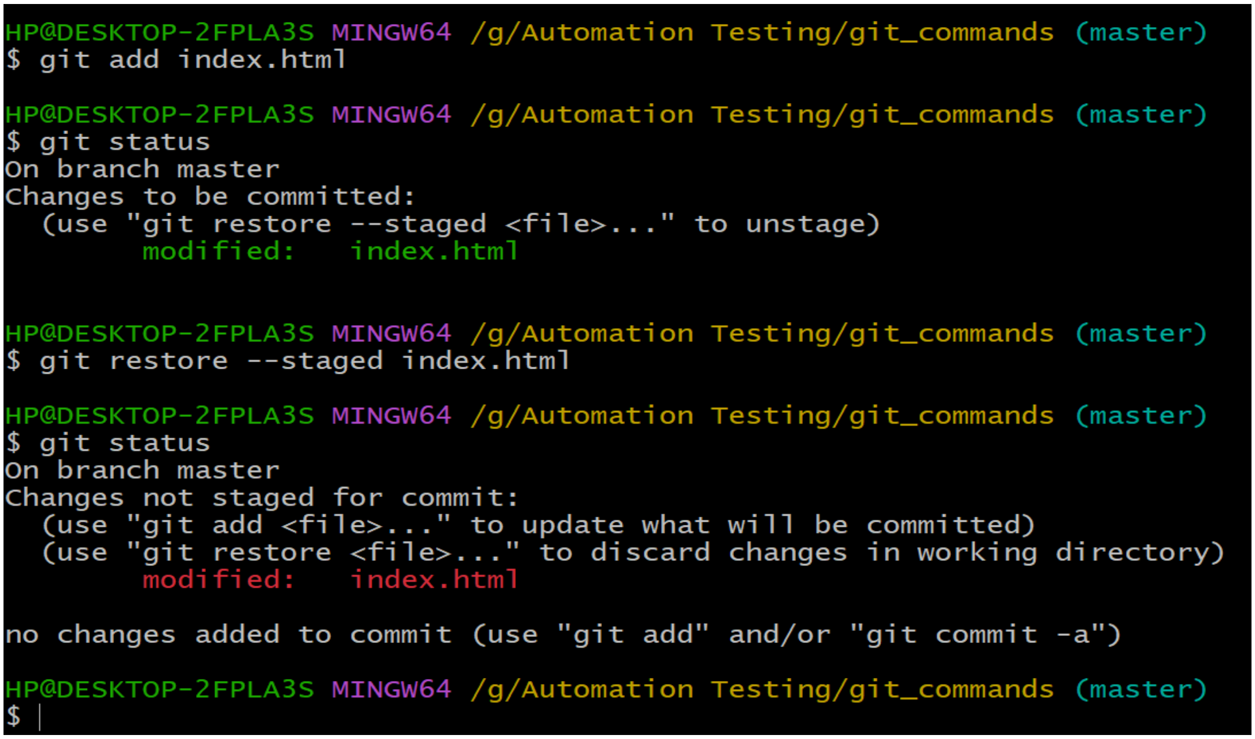
```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git diff head
warning: in the working copy of 'index.html', LF will be replaced by CRLF the next time Git touches it
diff --git a/index.html b/index.html
index 9ba1822..ed3ed2c 100644
--- a/index.html
+++ b/index.html
@@ -1 +1,5 @@
 Hello i am Aishu
+
+8:54 -- new data
+
+9:21 -- latest data
```

- Git restore



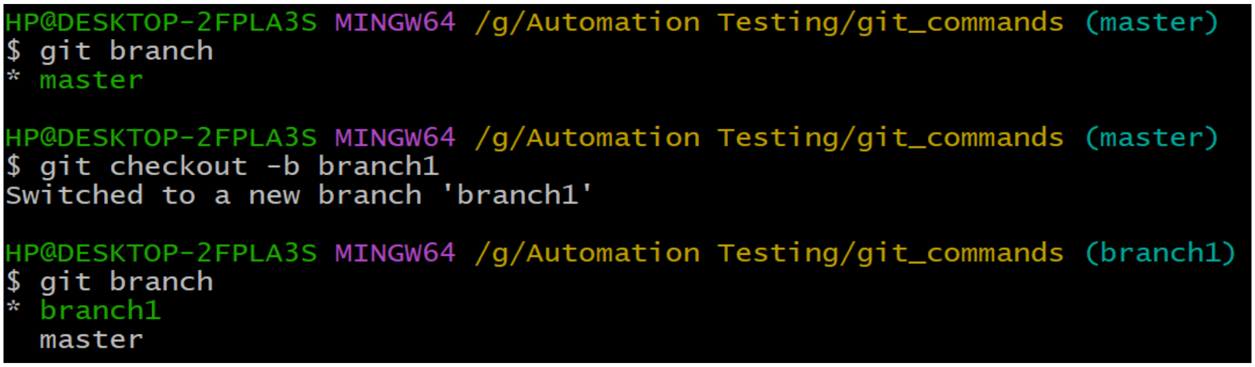
```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")

HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git restore index.html
```

- Git restore –staged



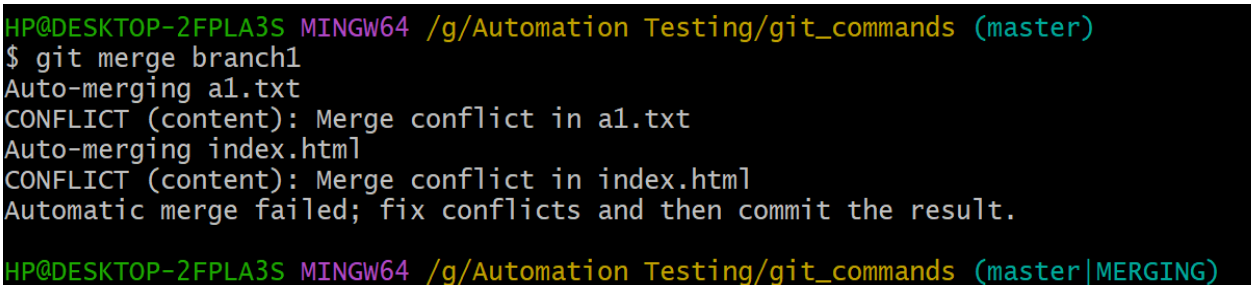
```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git add index.html

HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   index.html


HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git restore --staged index.html

HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")

HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ 
```

- Git branching
  - Git checkout -b branch1
  - Git branch



```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git branch
* master

HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git checkout -b branch1
Switched to a new branch 'branch1'

HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (branch1)
$ git branch
* branch1
  master
```

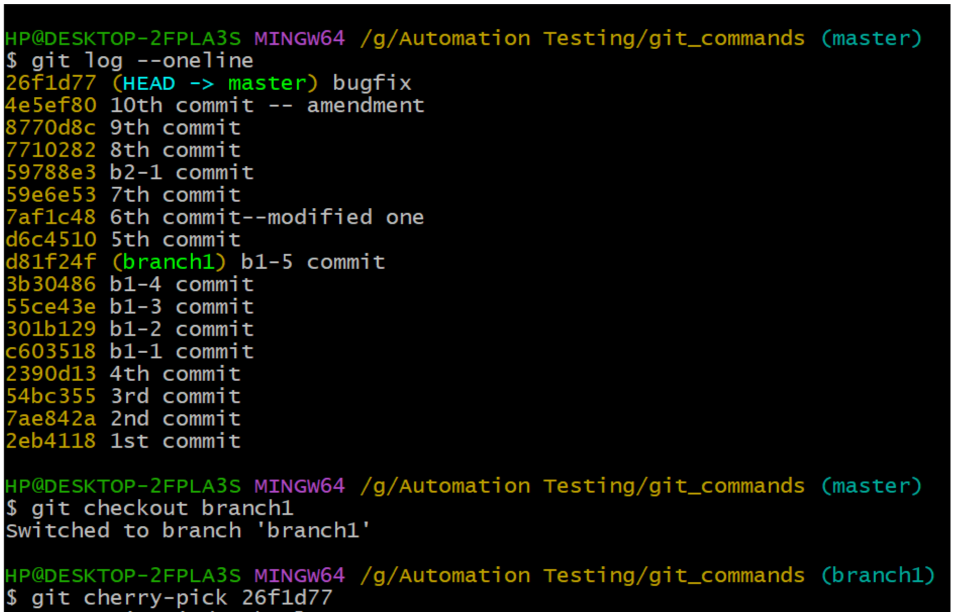
- Git rebase branch-name - rebase doesn’t add the extra commit.
- Git merge branch-name - merge the desired branch into the current branch & add one extra merge commit



```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git merge branch1
Auto-merging a1.txt
CONFLICT (content): Merge conflict in a1.txt
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.

HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master|MERGING)
```

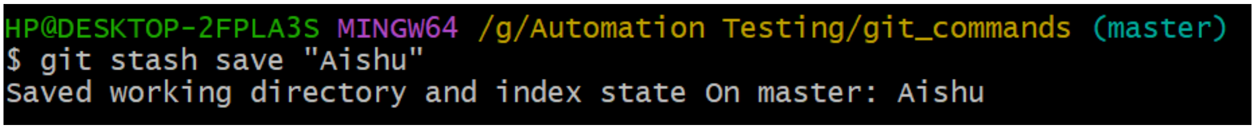
- Git commit --amend : – chnage the latest commit

- Git cherry-pick hash – apply specific commit from one ranch o another branch

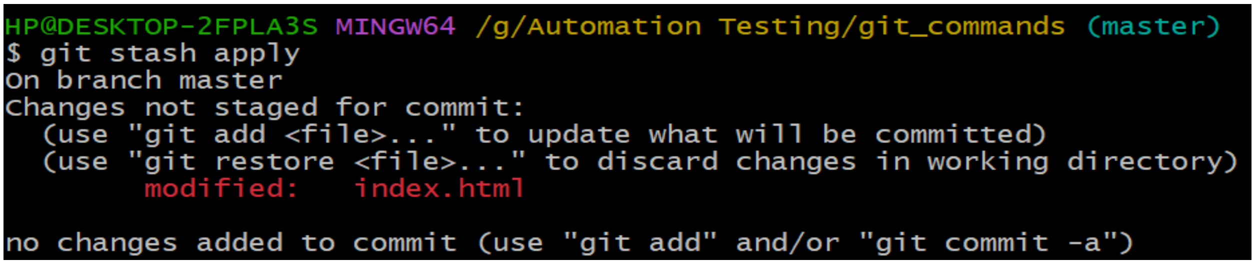


```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git log --oneline
26f1d77 (HEAD -> master) bugfix
4e5ef80 10th commit -- amendment
8770d8c 9th commit
7710282 8th commit
59788e3 b2-1 commit
59e6e53 7th commit
7af1c48 6th commit--modified one
d6c4510 5th commit
d81f24f (branch1) b1-5 commit
3b30486 b1-4 commit
55ce43e b1-3 commit
301b129 b1-2 commit
c603518 b1-1 commit
2390d13 4th commit
54bc355 3rd commit
7ae842a 2nd commit
2eb4118 1st commit

HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git checkout branch1
Switched to branch 'branch1'

HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (branch1)
$ git cherry-pick 26f1d77
```

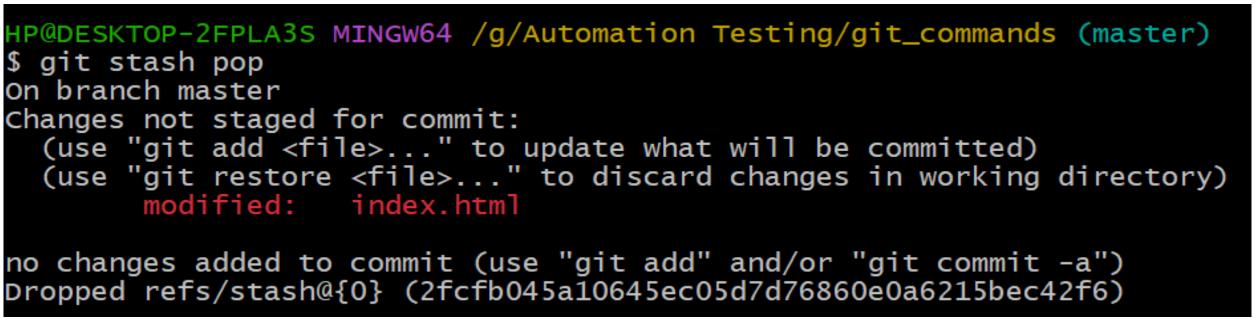
- Git stash save “stash-name”– save the changes in the working directory to stash
- Git stash pop – remove & apply the stash from the stash list
- Git stash apply - apply the stash without removing it from the stash list
- Git stash list – list the contents in the stash
- git stash apply stash@{1} – apply the specific stash



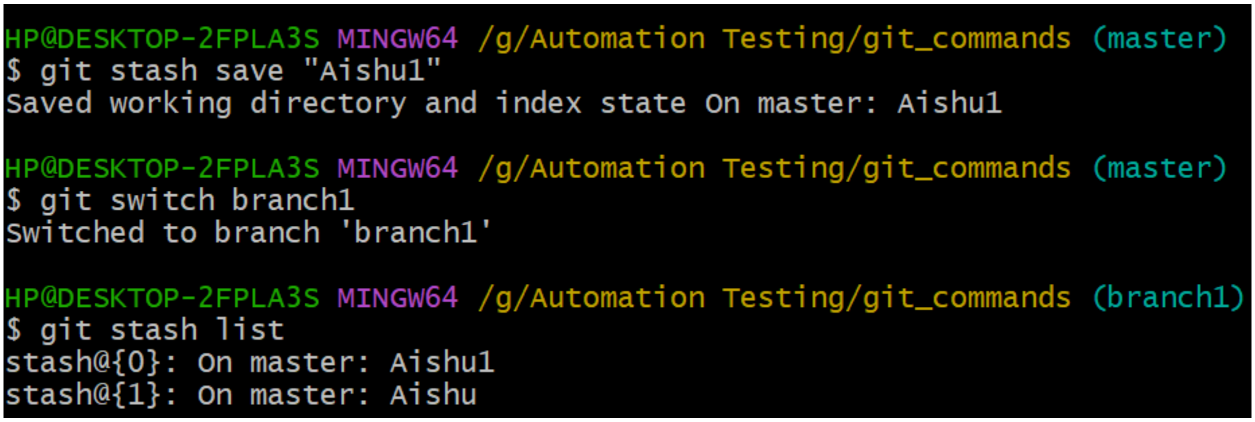
```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git stash save "Aishu"
Saved working directory and index state On master: Aishu
```



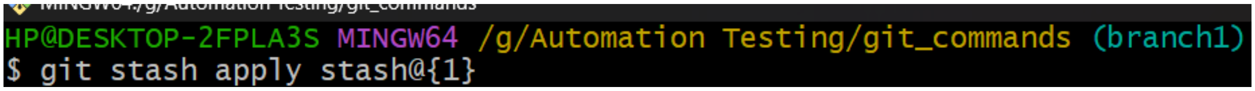
```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git stash apply
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")
```



```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git stash pop
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (2fcfb045a10645ec05d7d76860e0a6215bec42f6)
```



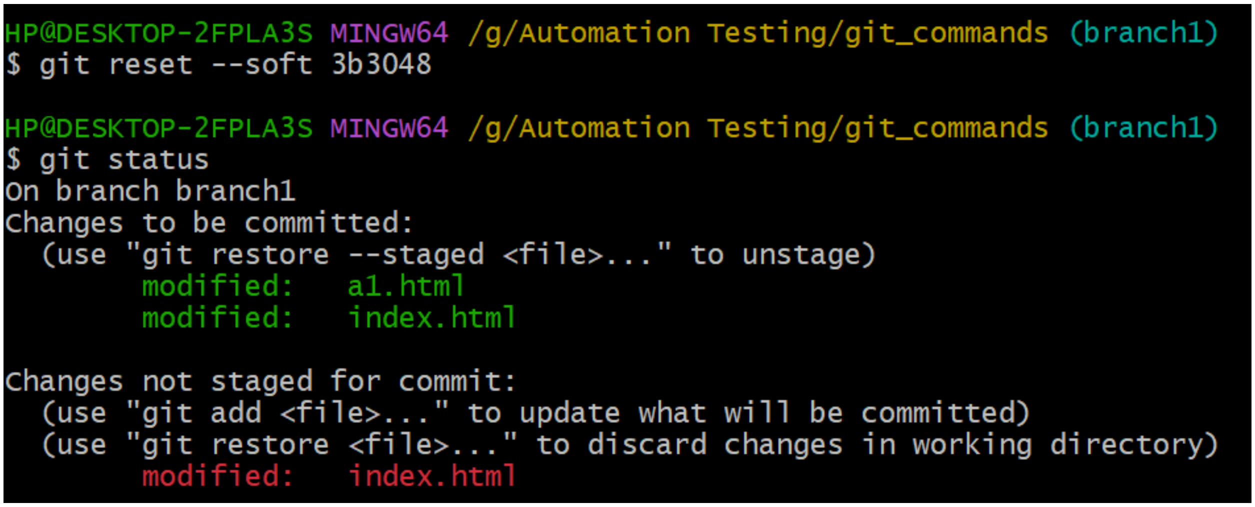
```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git stash save "Aishu1"
Saved working directory and index state On master: Aishu1

HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (master)
$ git switch branch1
Switched to branch 'branch1'

HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (branch1)
$ git stash list
stash@{0}: On master: Aishu1
stash@{1}: On master: Aishu
```



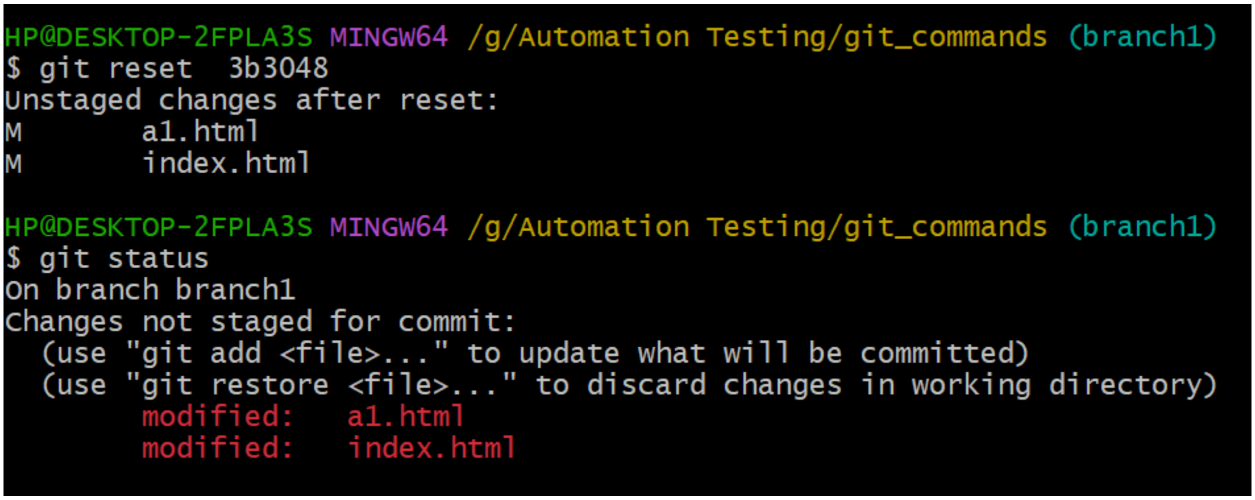
```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (branch1)
$ git stash apply stash@{1}
```

- Git reset :- default flag is mixed

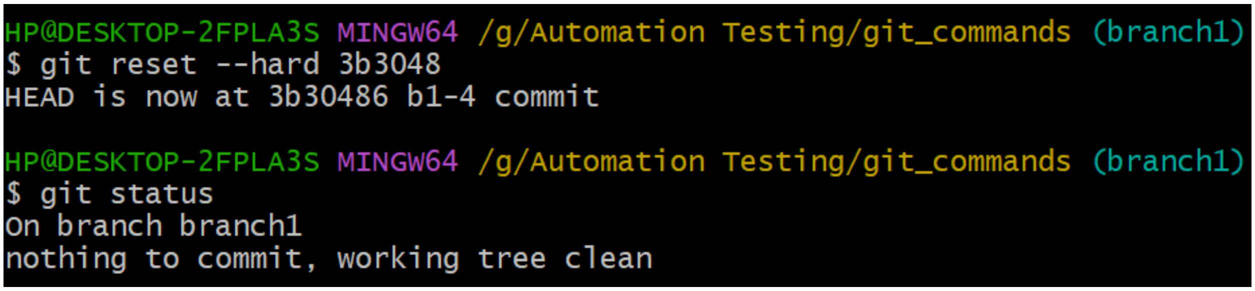
- Soft : Moves the branch pointer to the specified commit.Leaves the changes staged (changes will not be lost). keeps the changes in the working directory as it is .
- Mixed : Moves the branch pointer to the specified commit.Unstages the changes (changes are not lost). keeps the changes in the working directory as it is .
- Hard : Moves the branch pointer to the specified commit.Discards all changes (working directory and staging area).



```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (branch1)
$ git reset --soft 3b3048

HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (branch1)
$ git status
On branch branch1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   a1.html
        modified:   index.html

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html
```



```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (branch1)
$ git reset  3b3048
Unstaged changes after reset:
M       a1.html
M       index.html

HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (branch1)
$ git status
On branch branch1
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   a1.html
        modified:   index.html
```



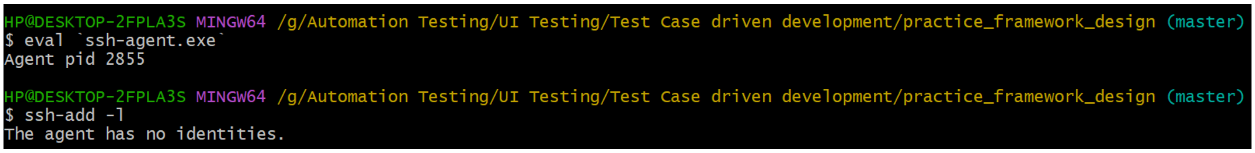
```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (branch1)
$ git reset --hard 3b3048
HEAD is now at 3b30486 b1-4 commit

HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/git_commands (branch1)
$ git status
On branch branch1
nothing to commit, working tree clean
```

- Git reflog - This command will show a list of recent actions and their corresponding commit hashes and references. It's a useful tool for tracking changes and can be particularly handy for recovering lost commits or branches.

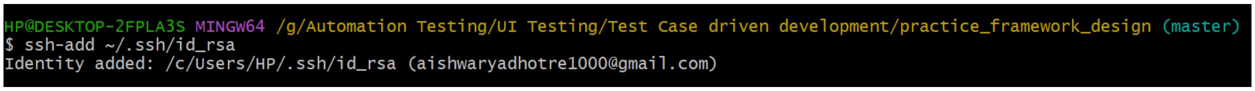
Push changes to remote repository:-

1. Create github repo
2. Generate ssh keys
3. Add public key to github
4. Initialize the git in the working repository

- Start ssh-agent.exe service
- Add private key to ssh-agent

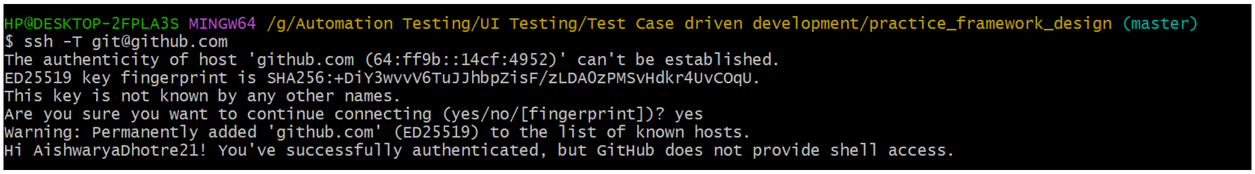


```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/UI Testing/Test Case driven development/practice_framework_design (master)
$ eval `ssh-agent.exe`
Agent pid 2855

HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/UI Testing/Test Case driven development/practice_framework_design (master)
$ ssh-add -l
The agent has no identities.
```



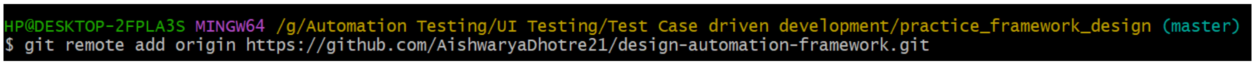
```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/UI Testing/Test Case driven development/practice_framework_design (master)
$ ssh-add ~/.ssh/id_rsa
Identity added: /c/Users/HP/.ssh/id_rsa (aishwaryadhotre1000@gmail.com)
```

- Authenticate with github

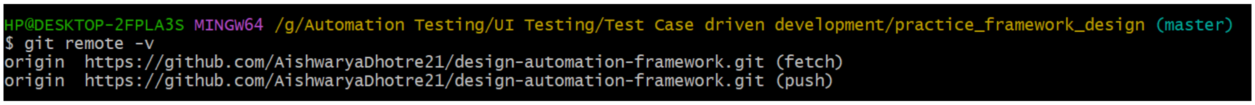


```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/UI Testing/Test Case driven development/practice_framework_design (master)
$ ssh -T git@github.com
The authenticity of host 'github.com (64:ff9b::14cf:4952)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
Hi AishwaryaDhotre21! You've successfully authenticated, but GitHub does not provide shell access.
```

Add remote url

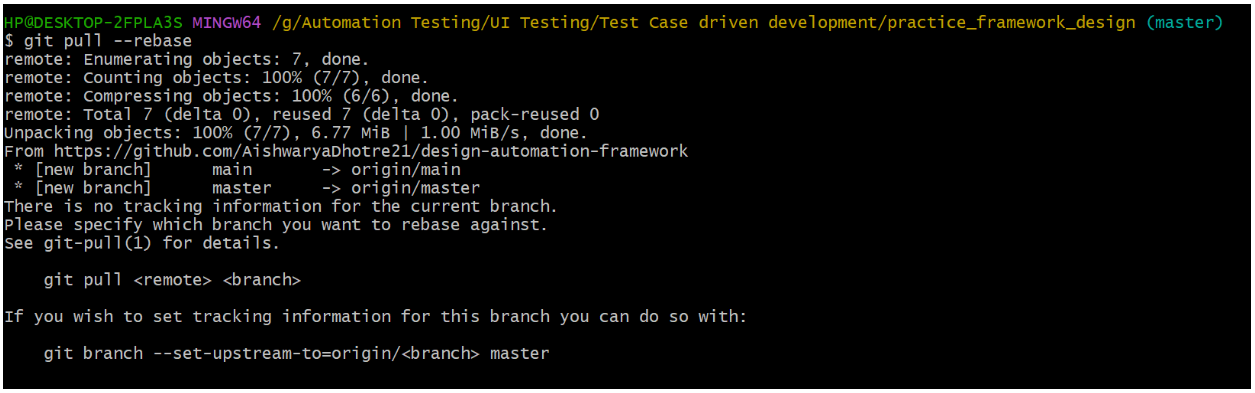


```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/UI Testing/Test Case driven development/practice_framework_design (master)
$ git remote add origin https://github.com/AishwaryaDhotre21/design-automation-framework.git
```

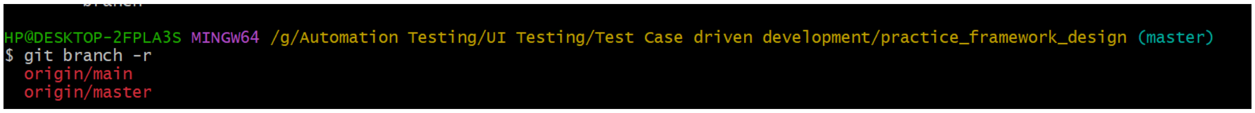


```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/UI Testing/Test Case driven development/practice_framework_design (master)
$ git remote -v
origin  https://github.com/AishwaryaDhotre21/design-automation-framework.git (fetch)
origin  https://github.com/AishwaryaDhotre21/design-automation-framework.git (push)
```

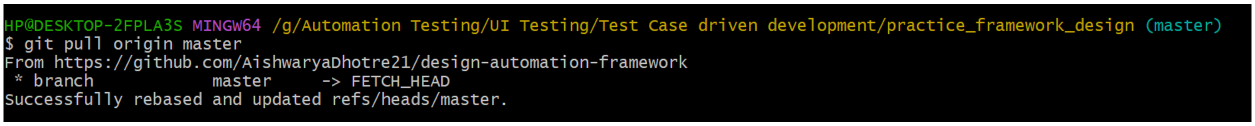
Rebase the remote branch on local branch



```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/UI Testing/Test Case driven development/practice_framework_design (master)
$ git pull --rebase
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 7 (delta 0), reused 7 (delta 0), pack-reused 0
Unpacking objects: 100% (7/7), 6.77 MiB | 1.00 MiB/s, done.
From https://github.com/AishwaryaDhotre21/design-automation-framework
 * [new branch]      main       -> origin/main
 * [new branch]      master     -> origin/master
There is no tracking information for the current branch.
Please specify which branch you want to rebase against.
See git-pull(1) for details.

    git pull <remote> <branch>

If you wish to set tracking information for this branch you can do so with:

    git branch --set-upstream-to=origin/<branch> master
```

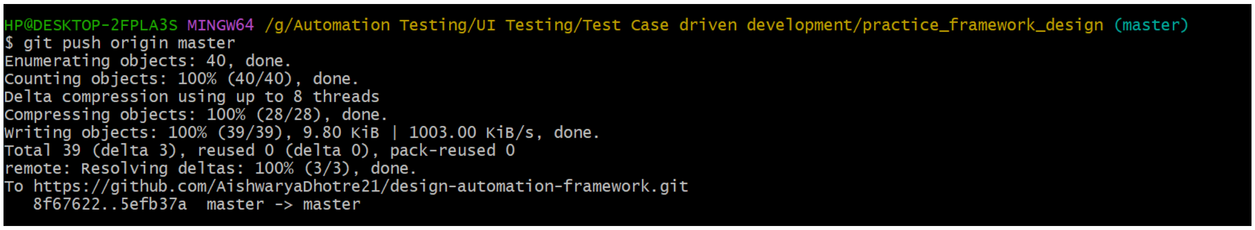


```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/UI Testing/Test Case driven development/practice_framework_design (master)
$ git branch -r
  origin/main
  origin/master
```



```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/UI Testing/Test Case driven development/practice_framework_design (master)
$ git pull origin master
From https://github.com/AishwaryaDhotre21/design-automation-framework
 * branch            master     -> FETCH_HEAD
Successfully rebased and updated refs/heads/master.
```

Push the code



```
HP@DESKTOP-2FPLA3S MINGW64 /g/Automation Testing/UI Testing/Test Case driven development/practice_framework_design (master)
$ git push origin master
Enumerating objects: 40, done.
Counting objects: 100% (40/40), done.
Delta compression using up to 8 threads
Compressing objects: 100% (28/28), done.
Writing objects: 100% (39/39), 9.80 KiB | 1003.00 KiB/s, done.
Total 39 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/AishwaryaDhotre21/design-automation-framework.git
   8f67622..5efb37a  master -> master
```