NEW Cheat Sheet

-git init

-git status

-git add .

-git commit –m “AnyComment”

-git status

-git remote add origin URL of repository with extension .git

-git remote –v

-git push origin master

-git push origin master --force

Only single file

$git add filename.text

$git commit –m “comment”

-git push origin master --force

To add multiples files in one go

-git add \*

$git commit –m “comment”

-git push origin master –force

1st time for reference issue

Git init

Git remote add origin url of your git with .git

Git remote // optional

Git push –u origin master // might give error for ref

Git add –A

Git commit –m “comments”

Git push –u origin master

Git push origin master - -force // optional

Git status

*If not uploaded in git then continue*

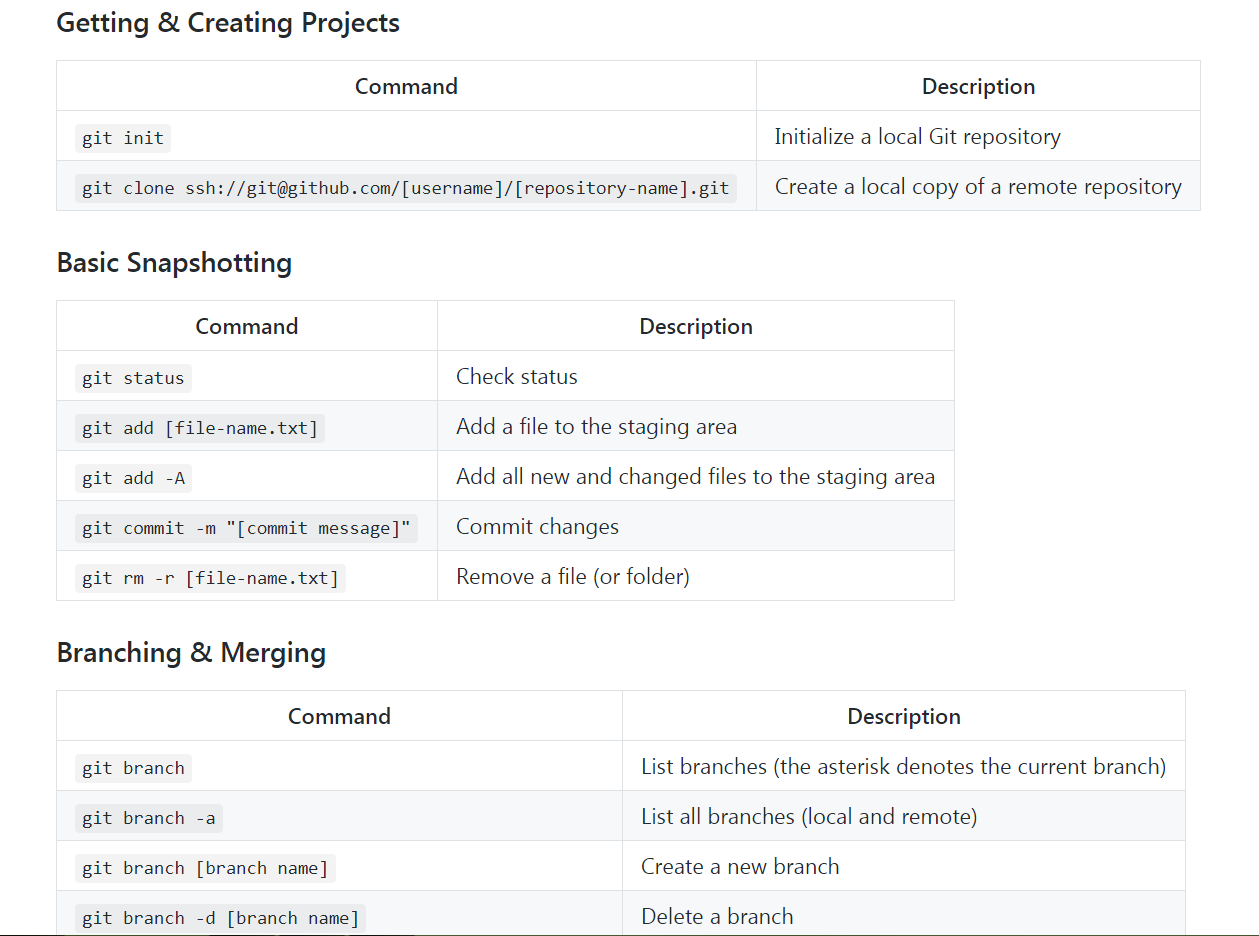
Git add .

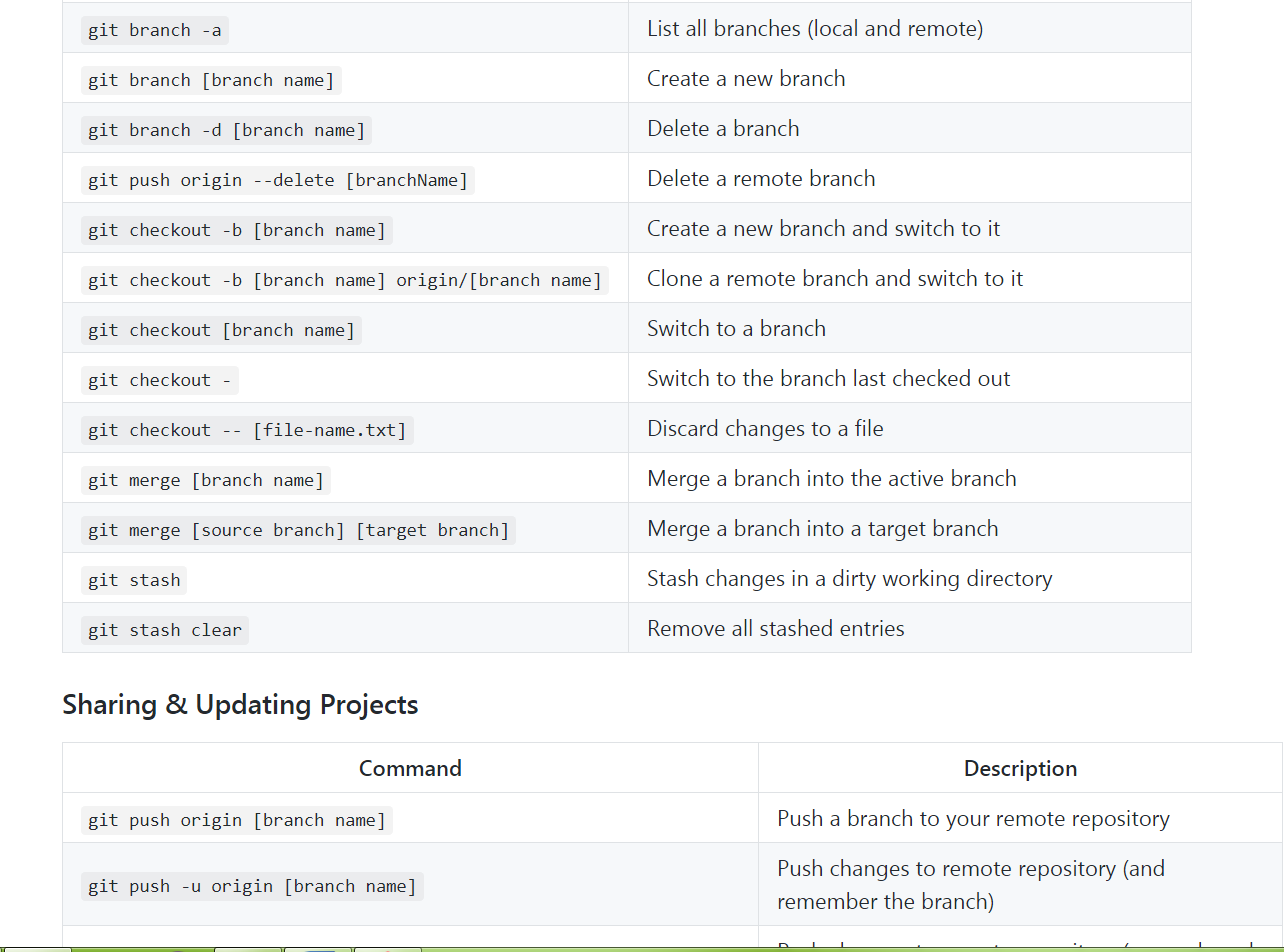
Git commit –m “comments”

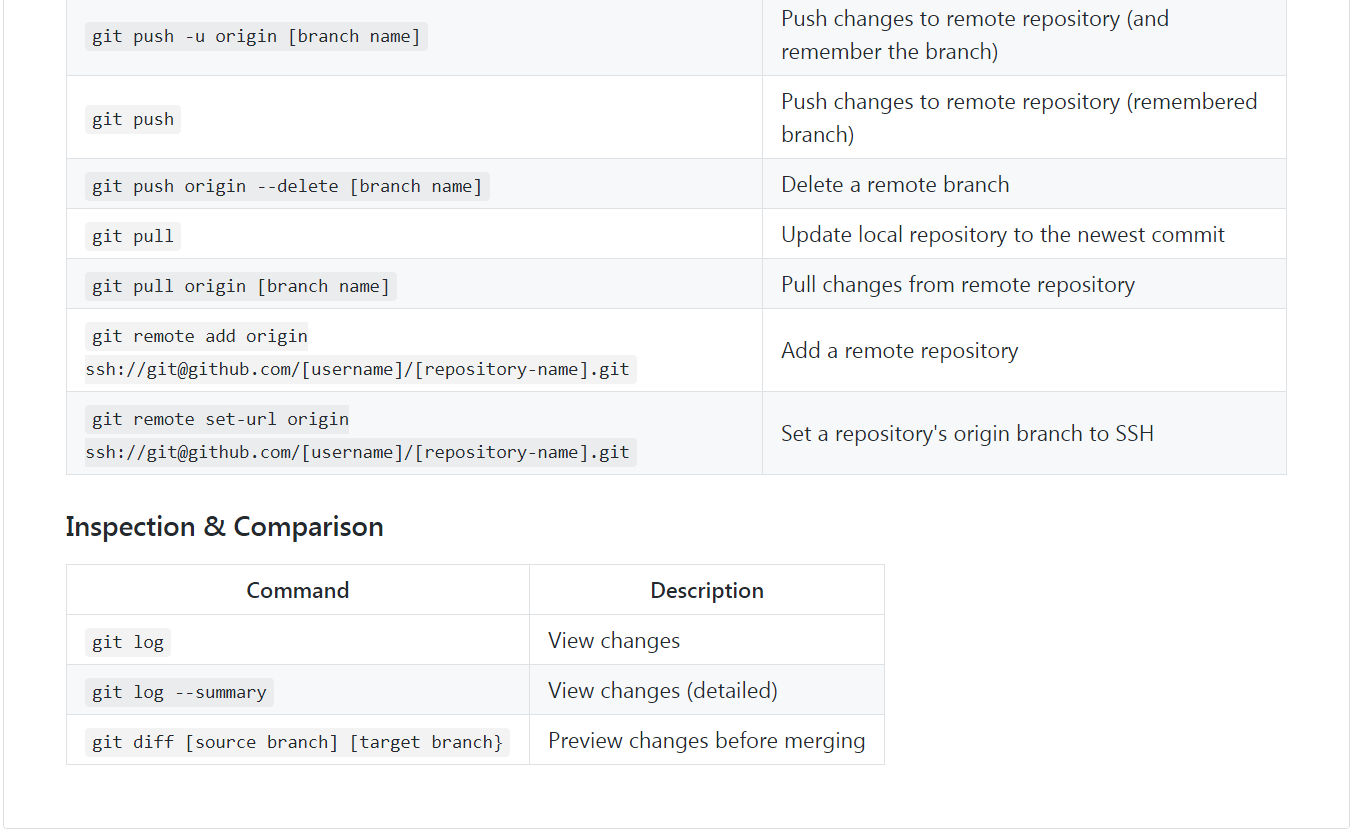
Git remote –v

Git push origin master

Git push origin master - -force







* **git config**

Sets configuration values for your user name, email, gpg key, preferred diff algorithm, file formats and more. Examples:

|  |  |
| --- | --- |
| 1  2 | git config --global user.name "My Name"  git config --global user.email "user@domain.com" |

* **git init**

Initializes a git repository – creates the initial *.git* directory in a new or in an existing project. Example:

|  |  |
| --- | --- |
| 1  2  3 | git init    Initialized empty Git repository in /home/username/GIT/.git/ |

* **git clone**

Creates a GIT repository copy from a remote source. Also adds the original location as a remote so you can fetch from it again and push to it if you have permissions. Example:

|  |  |
| --- | --- |
| 1 | git clone [git@github.com:user/test.git](mailto:git@github.com:user/test.git) |

* **git add**

Adds files changes in your working directory to your index. Example:

|  |  |
| --- | --- |
| 1 | git add . |

* **git rm**

Removes files from your index and your working directory so they will not be tracked. Example:

|  |  |
| --- | --- |
| 1 | git rm filename |

* **git commit**

Takes all of the changes written in the index, creates a new commit object pointing to it and sets the branch to point to that new commit. Examples:

|  |  |
| --- | --- |
| 1  2 | git commit -m ‘committing added changes’  git commit -a -m ‘committing all changes, equals to git add and git commit’ |

* **git status**

Shows you the status of files in the index versus the working directory. It will list out files that are untracked (only in your working directory), modified (tracked but not yet updated in your index), and staged (added to your index and ready for committing). Example:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | git status    # On branch master #  # Initial commit #  # Untracked files: #  # (use "git add <file>..." to include in what will be committed) #    README |

* **git branch**

Lists existing branches, including remote branches if ‘-a’ is provided. Creates a new branch if a branch name is provided. Example:

|  |  |
| --- | --- |
| 1 | git branch -a \* master remotes/origin/master |

* **git merge**

Merges one or more branches into your current branch and automatically creates a new commit if there are no conflicts. Example:

|  |  |
| --- | --- |
| 1 | git merge newbranchversion |

* **git reset**

Resets your index and working directory to the state of your last commit. Example:

|  |  |
| --- | --- |
| 1 | git reset --hard HEAD |

* **git tag**

Tags a specific commit with a simple, human readable handle that never moves. Example:

|  |  |
| --- | --- |
| 1 | git tag -a v1.0 -m 'this is version 1.0 tag' |

* **git pull**

Fetches the files from the remote repository and merges it with your local one. Example:

|  |  |
| --- | --- |
| 1 | git pull origin |

* **git push**

Pushes all the modified local objects to the remote repository and advances its branches. Example:

|  |  |
| --- | --- |
| 1 | git push origin master |

* **git remote**

Shows all the remote versions of your repository. Example:

|  |  |
| --- | --- |
| 1 | git remote origin |

* **git log**

Shows a listing of commits on a branch including the corresponding details. Example:

|  |  |
| --- | --- |
| 1  2  3 | git log commit    84f241e8a0d768fb37ff7ad40e294b61a99a0abe Author: User <user@domain.com> Date: Mon May 3 09:24:05 2010 +0300 first commit |

* **git diff**

Generates patch files or statistics of differences between paths or files in your git repository, or your index or your working directory. Example:

|  |  |
| --- | --- |
| 1 | git diff |

* **git archive**

Creates a tar or zip file including the contents of a single tree from your repository. Example:

|  |  |
| --- | --- |
| 1 | git archive --format=zip master^ README >file.zip |

* **git gc**

Garbage collector for your repository. Optimizes your repository. Should be run occasionally. Example:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | git gc    Counting objects: 7, done.  Delta compression using up to 2 threads.  Compressing objects: 100% (5/5), done.  Writing objects: 100% (7/7), done.  Total 7 (delta 1), reused 0 (delta 0) |

* **git fsck**

Does an integrity check of the Git file system, identifying corrupted objects. Example:

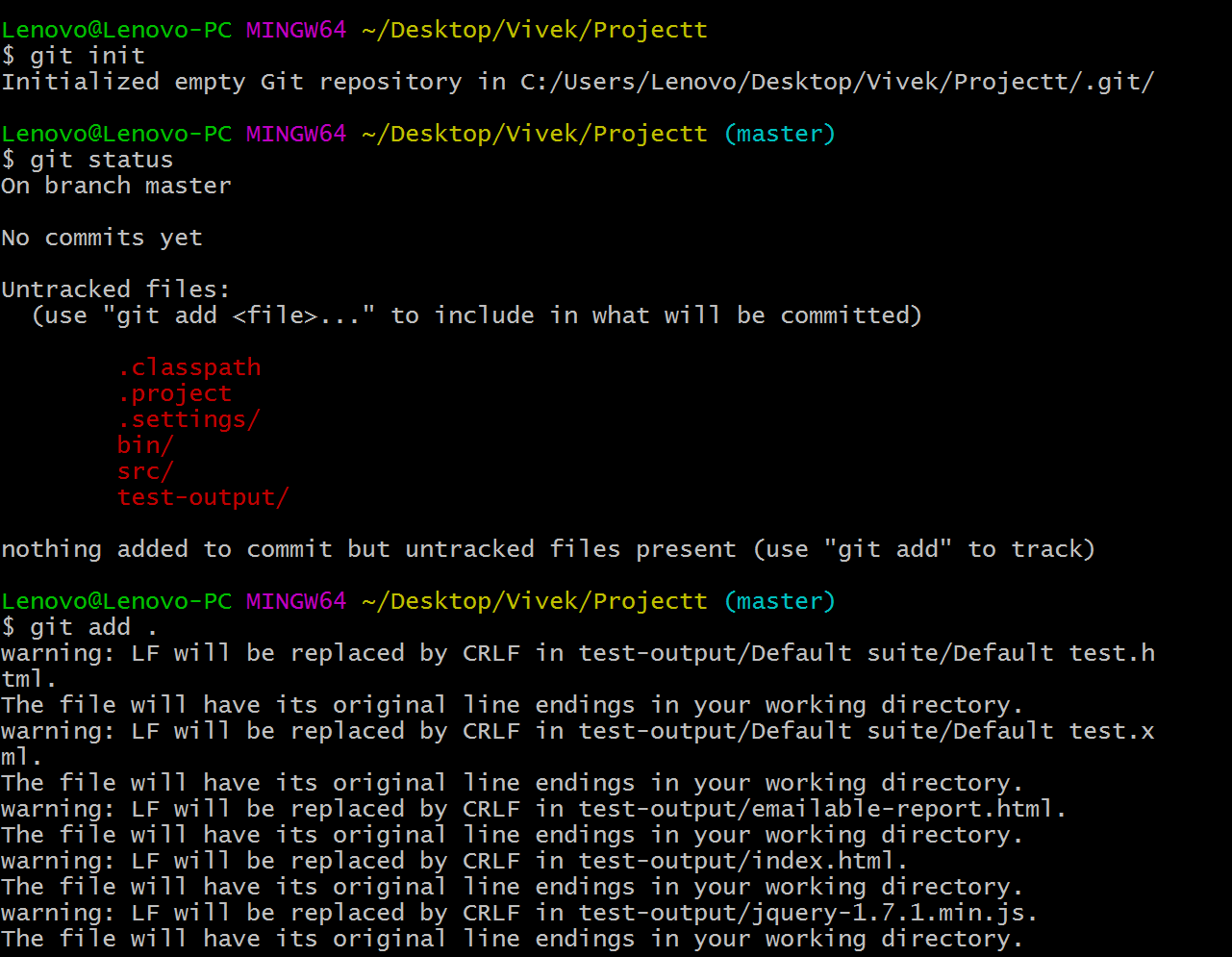
|  |  |
| --- | --- |
| 1 | git fsck |

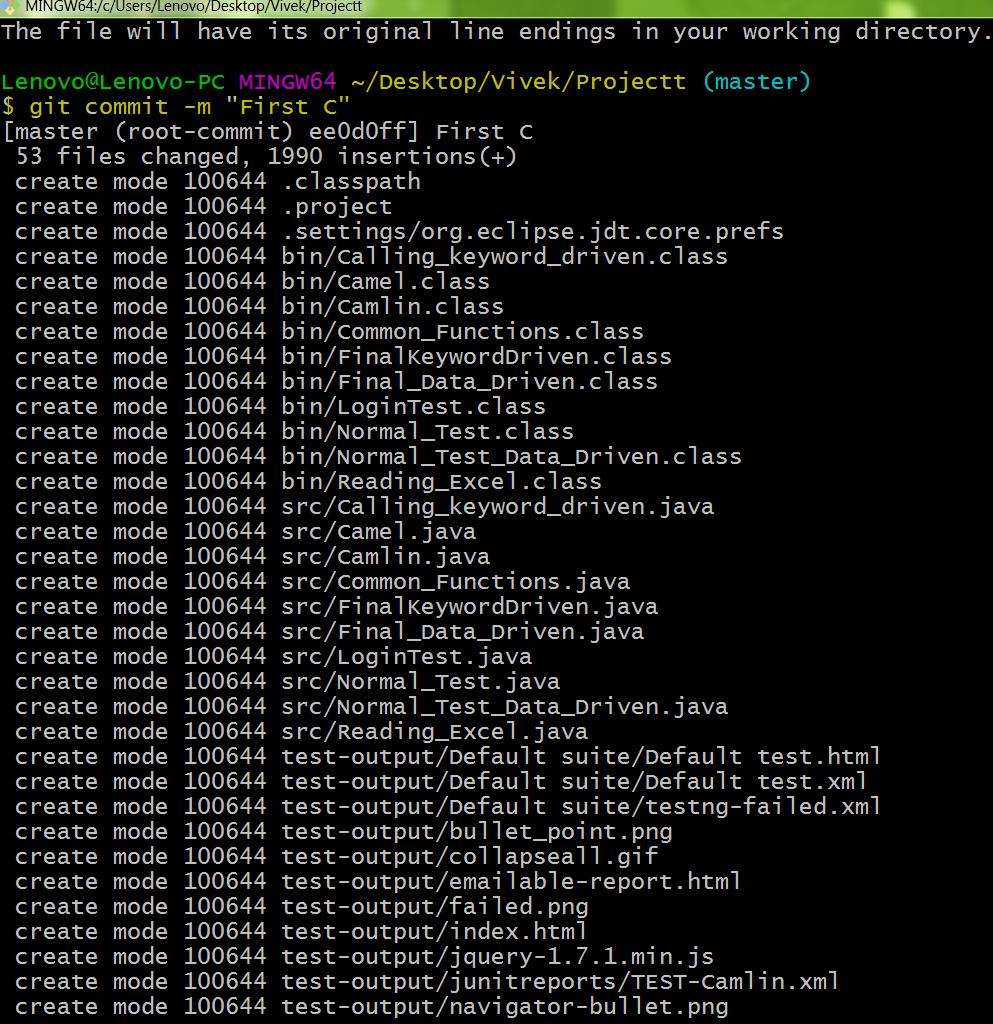
* **git prune**

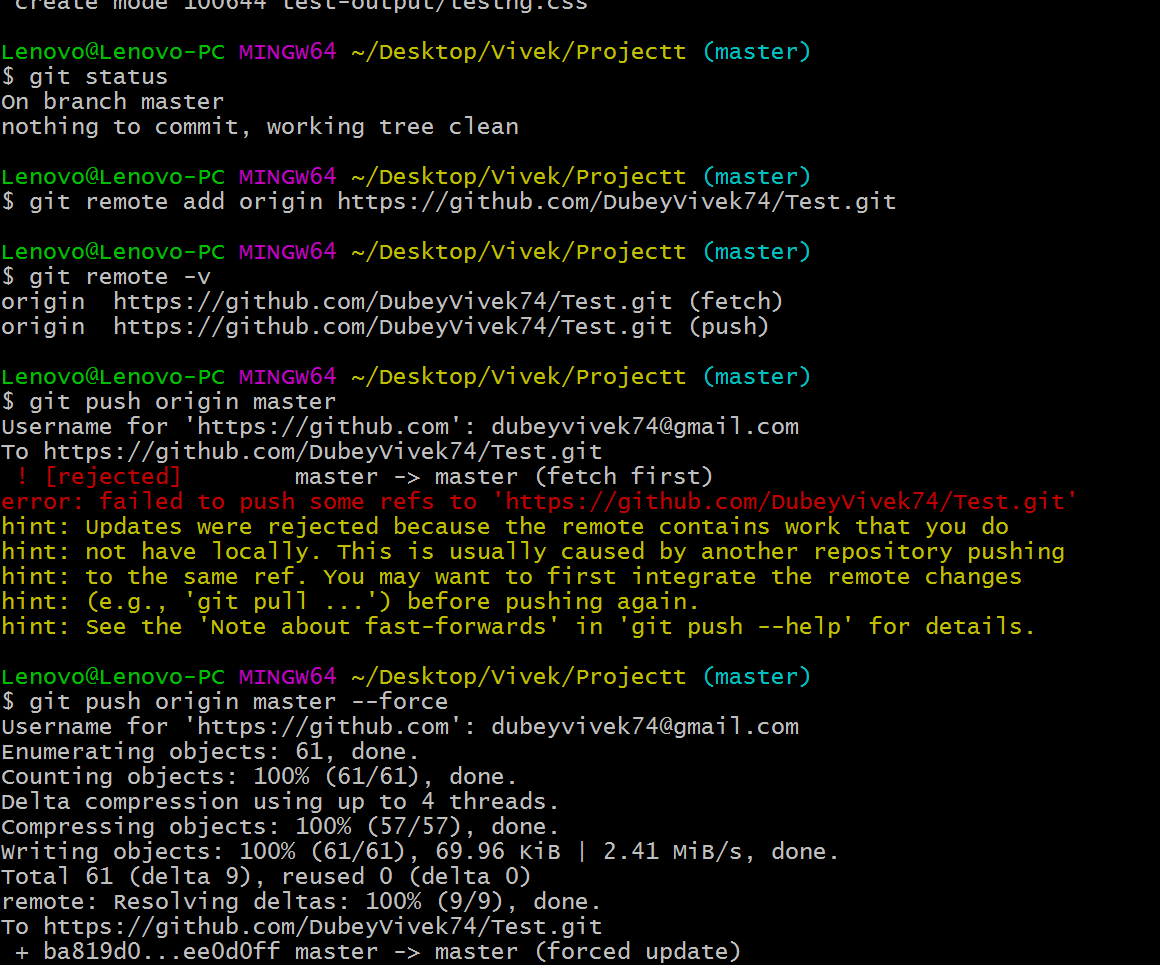
Removes objects that are no longer pointed to by any object in any reachable branch. Example:

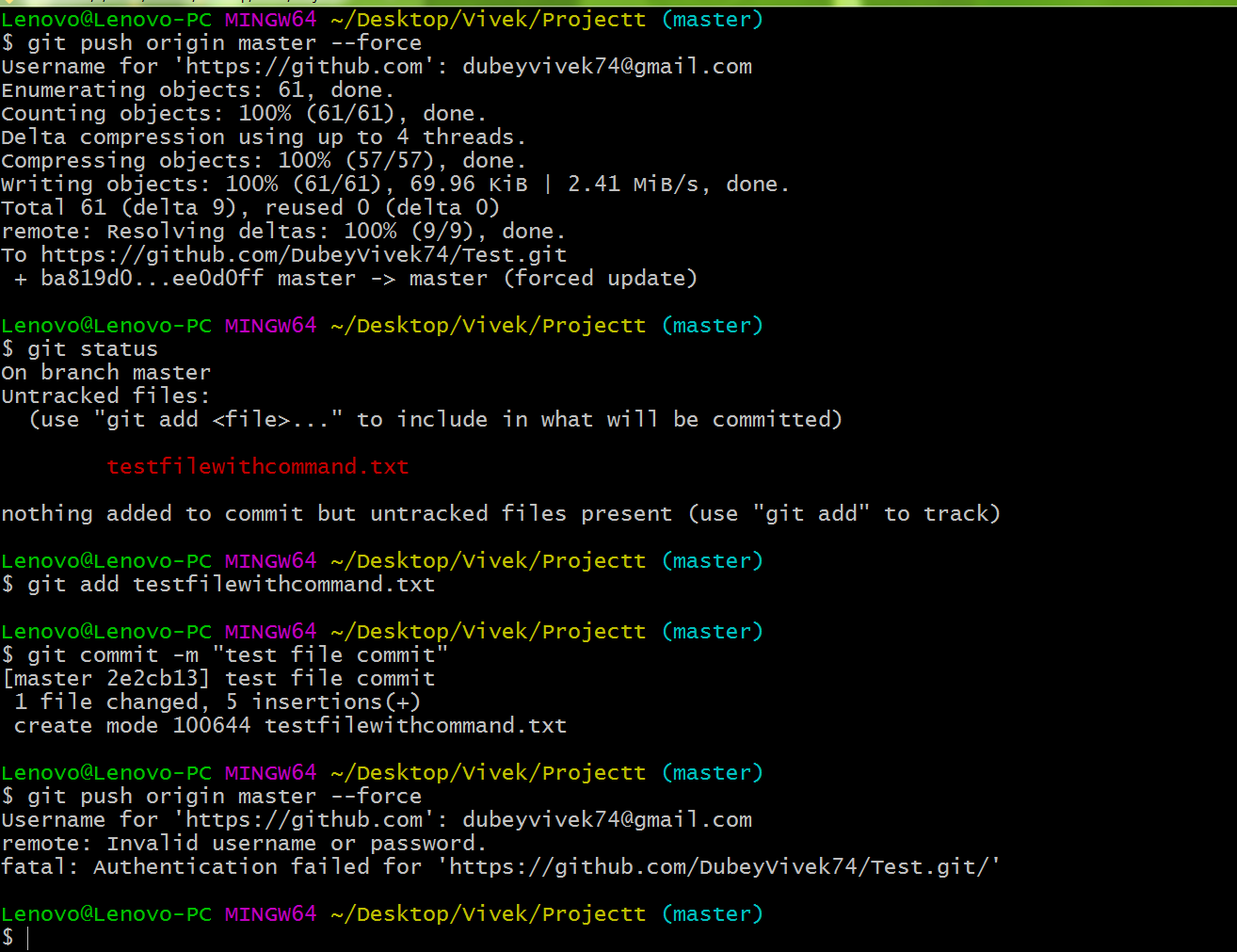
|  |  |
| --- | --- |
| 1 | git prune |

**Ignore screen shots just for demo**









----TRIAL @

2

