Sets configuration values for your user name, email as follows One time only

```
#git config --global user.name "DevOps Training Bangalore"
#git config --global user.email "devopstrainingblr@gmail.com"
```

Checking for settings

```
#git config --list
```

You can also check what Git thinks a specific key’s value is by typing git config <key>:

```
#git config user.name
```

```
#git config --global core.editor "nano"  (GNU nano 2.0)
```

(OR)

```
#git config --global core.editor "vim"  (VIM)
```

Task 1: Create the git local repository in local machine (Laptop/Deskto), add one file (bhaska1.txt) and update that file, create the github remote repository (https://github.com) and move the local code to github repository.

Go the directory where you want to create the git repository.

```
# mkdir git-practice-commands
#cd git-practice-commands
```

```
#git init
```

: Create a local Git repository.

```
Initialized empty Git repository in /Users/BhaskarReddy/git/git-practice-commands/.git/
```

```
#git status
```

: Gives the status of your untracked files.

```
On branch master

Initial commit


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

```
#touch DbConnect.java
#git status
```

```
On branch master

Initial commit

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

    bhaskar1.txt

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

#vim DbConnect.java
#git add DbConnect.java: Add the files(here bhaskar1.txt) into your new local repository. This stages them for the first commit.
#git status

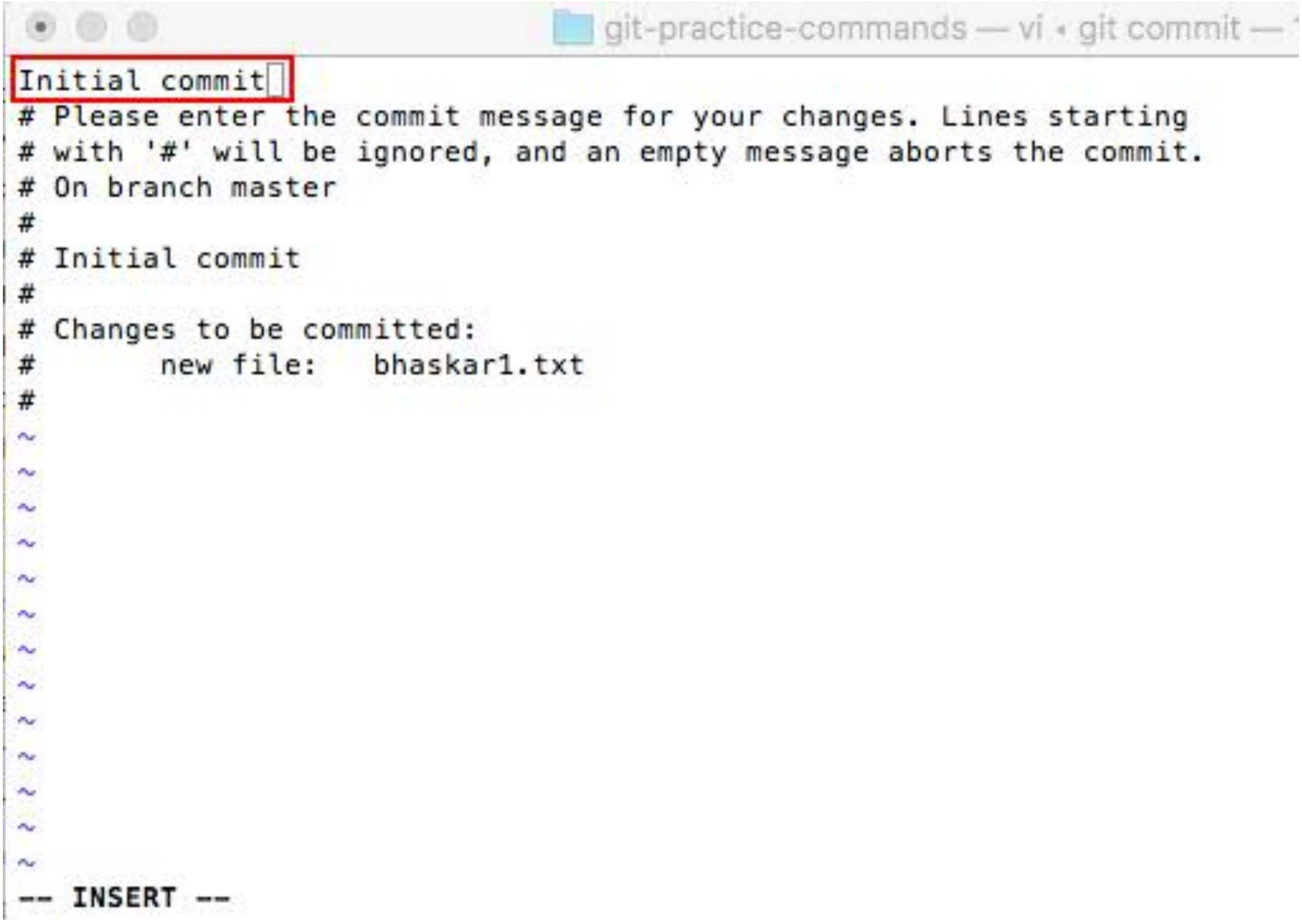
```
On branch master

Initial commit

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

    new file:   bhaskar1.txt
```

# git commit : Commit the files that you've staged in your local repository.



```
Initial commit
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# On branch master
#
# Initial commit
#
# Changes to be committed:
#       new file:   bhaskar1.txt
#
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

type i then put the comment and press esc key and type :wq and click on enter key.

#git status

```
On branch master

nothing to commit, working tree clean
```

- Open the file (DbConnect.java) and update with some text.

#git status

```
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working
directory)

    modified:   Dbconnect.java


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

#git commit -a -m "Updated DbConnect.java file"

```
[master 7f795a7] Updated DbConnect.java file
 1 file changed, 1 insertion(+)
```

- Create the repository in github as follows.

Login into github (http://github.com)
On right side top corner click on **"+"** symbol and click on **"New repository"** and give the Repository name and click on Create repository.

#git remote add origin git@github.com:bhaskar0504/test.git : Adding the URL for the remote repository where your local repository will be pushed.

# git remote -v

#git push -u origin master : Push the changes in your local repository to GitHub remote repository. (Here push is the git command , origin is the remote name and master is the branch name)

```
Counting objects: 6, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (6/6), 479 bytes | 0 bytes/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To git@github.com:bhaskar0504/test.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
```

#git status

```
 On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working tree clean
```

#git remote show origin : It will give the information on a particular remote (here origin is the remote name)

# git remote remove origin : It will remove the remote origins.

# git remote -v

Task 2: Create the git local repository in local machine (Laptop/Deskto), create the another branch (bugfix), merge with master branch.

#mkdir branchmerge
#cd branchmerge
#git init (By default it will create the master branch)

#touch Bhaskar.txt
Update this file like change 1 – master branch
#vim Bhaskar.txt
# git add .
# git commit -a -m "initial commit"
#git branch : It gives the branch names in current repository.
#git branch bugfix : It will create the bugfix branch in local git repository.

#git branch -v: It will display the branches in git repo.

```
bugfix 87226db initial commit
* master 87226db initial commit
```

Note: Here * indicate currently in use branch.
# git checkout bugfix : Switch to bugfix branch.

```
Switched to branch 'bugfix'
```

Update the Bhaskar.txt like change 2 – bugfix branch
# git add .
# git commit -a -m "bugfix commit"

# git checkout master : Switch to master branch.

```
Switched to branch 'master'
```

Updat the Bhaskar.txt like change 3 – master branch
# git add .
# git commit -a -m "master commit"

# git checkout bugfix : Switch to bugfix branch.

```
Switched to branch 'bugfix'
```

Check the file and see the contents in file.

#git checkout master

#git diff master bugfix

#git merge bugfix
Fix the conflicts
#git add .
#git commit –m "merging"
#git remote add origin **<<Git Remote Repo>>**
#git push --all origin

#git branch -d bugfix

#git log

Mirroring a repository

To make an exact duplicate, you need to perform both a bare-clone and a mirror-push.
Open up the command line, and type these commands:

```
 Create one directory
#mkdir codebackup
#cd codebackup

# Make a bare clone of the repository
#git clone --bare https://github.com/exampleuser/old-repository.git

git push --mirror https://github.com/exampleuser/new-repository.git
# Mirror-push to the new repository

# Remove our temporary local repository
rm -rf codebackup
```

Steps for Code Checkout into local from Repository

```
========================================
```

Go to the directory where we need to commit the code/checkout the code

```
cd C:\BhaskarReddyL\JavaWorkspace\MTWorkSpace
```

Get the code from Git Repository as follows.

```
git clone <<GitHub URL>>
```

Git copying one repository to another empty one

In order to make an exact duplicate, you need to perform both a bare-clone and a mirror-push:

Step 1) Open terminal or CMD prompt and enter the below command.

git clone --bare << Git Repo URL>>

Once you enter that git command, it will ask you to enter the username(bhlaccha) and password (Personal access tokens)

Step 2) Once we downloaded the code into our local machine, execute the below command to move the code to new repository.

git push --mirror <<Git Repo URL>>

Git Commands

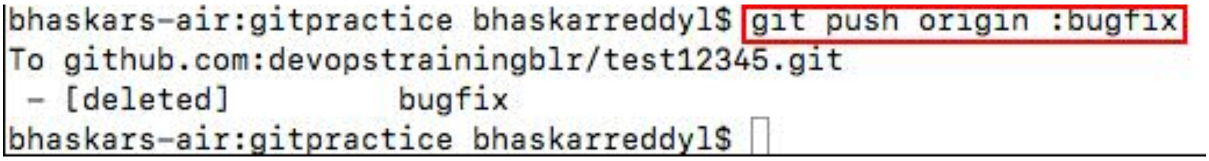
git branch : It will displays the branch names on your repository.

git branch <<Branch Name>> : It will create the local branch in the repository.

Ex: git branch test

git branch -D test : It will delete the local branch in the repository.

git push origin : bugfix : It will delete a remote branch in the repository.



```
bhaskars-air:gitpractice bhaskarreddyl$ git push origin :bugfix
To github.com:devopstrainingblr/test12345.git
 - [deleted]           bugfix
bhaskars-air:gitpractice bhaskarreddyl$ 
```

git checkout -b <<Branch name>> : It will create the branch name and will switch.

git checkout <<Branch name>> : This will switch the branch.

Ex: git checkout test

git config http.sslVerify false : To disable SSL verification for that singular repository

git config --global http.sslVerify false : To disable the SSL verification for Globally (For all repositories) --> Not suggested way

git clone <<Git URL>> : To get the code from repository into your local machine.

git log : It will display the commit history.

git log -p -2 : which shows the difference introduced in each commit. You can also use -2, which limits the output to only the last two entries:

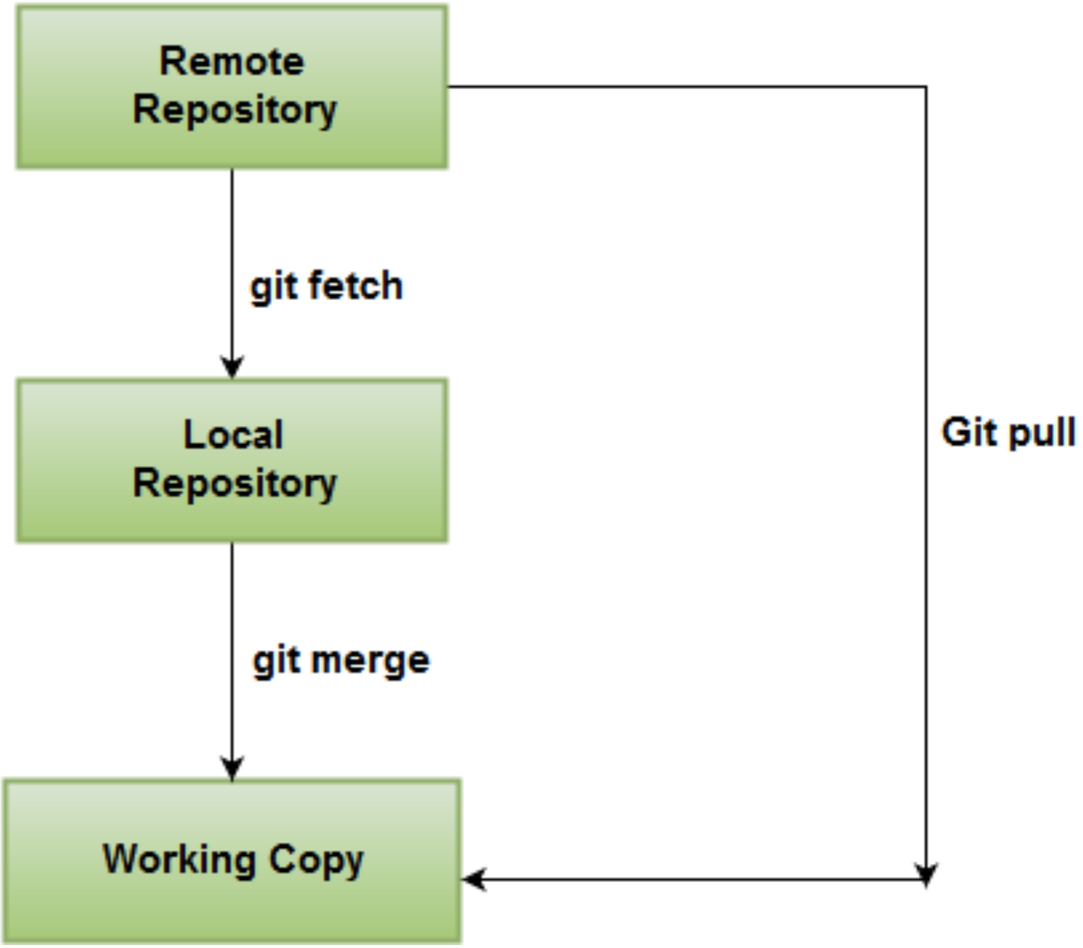
git log --stat : If you want to see some abbreviated stats for each commit, you can use the --stat option.

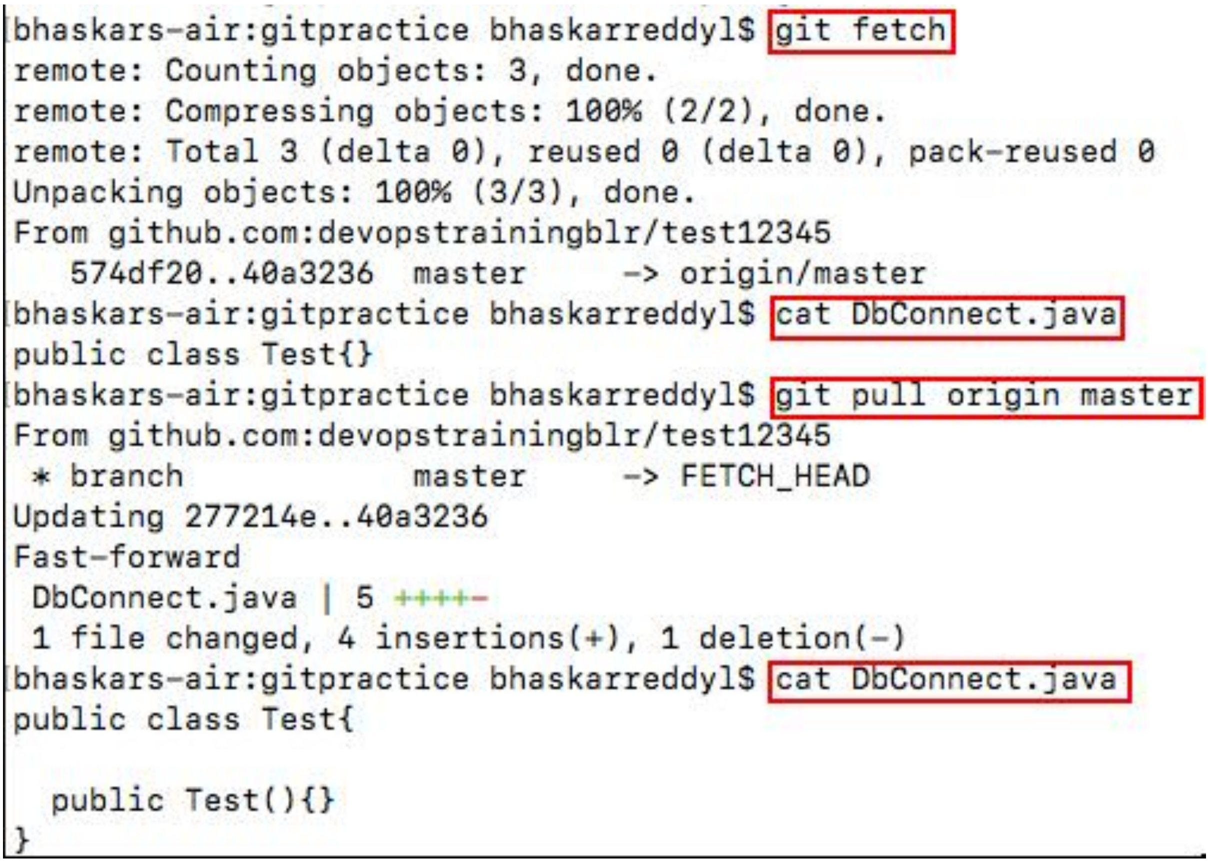
git rm : Removes files from your index and your working directory so they will not be tracked.

What is the difference between git fetch and get pull?

Ans) git fetch : It will get the update from git remote repo and will update your local repo. But it will not merge with Local working copy.
git fetch : It will get the update from git remote repo and will update your local repo as well it will merge with Local working copy also.

So **git pull = git fetch + git merge origin/master**

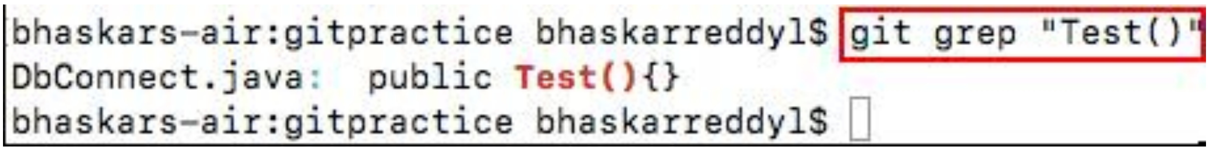




```
[bhaskars-air:gitpractice bhaskarreddyl$ git fetch
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From github.com:devopstrainingblr/test12345
   574df20..40a3236  master     -> origin/master
[bhaskars-air:gitpractice bhaskarreddyl$ cat DbConnect.java
public class Test{}
[bhaskars-air:gitpractice bhaskarreddyl$ git pull origin master
From github.com:devopstrainingblr/test12345
 * branch            master     -> FETCH_HEAD
Updating 277214e..40a3236
Fast-forward
 DbConnect.java | 5 ++++-
 1 file changed, 4 insertions(+), 1 deletion(-)
[bhaskars-air:gitpractice bhaskarreddyl$ cat DbConnect.java
public class Test{

  public Test(){}
}
```

# git grep "Test()" : Search the working directory for Test()



```
bhaskars-air:gitpractice bhaskarreddyl$ git grep "Test()"
DbConnect.java:   public Test(){}
bhaskars-air:gitpractice bhaskarreddyl$
```

Resources:
----
https://github.com/
https://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup
https://www.atlassian.com/git/tutorials/comparing-workflows/

https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging
http://www.vogella.com/tutorials/Git/article.html
https://help.github.com/articles/duplicating-a-repository/