5/15/2018

|  |
| --- |
| Git & GitHub Vibranarayanan |



|  |  |
| --- | --- |
| GIT & Github | Installation and POC on Windows |

Table of Contents

[Daily Topics tracking 4](#_Toc514242775)

[Git Installation 4](#_Toc514242776)

[Git download 4](#_Toc514242777)

[Git Installation 5](#_Toc514242778)

[Select Installation Path 5](#_Toc514242779)

[Select Component for Installation 5](#_Toc514242780)

[Setting Path Environment 6](#_Toc514242781)

[Other Configuration set-up during installation 6](#_Toc514242782)

[Finish Installation 7](#_Toc514242783)

[Verify git installation 7](#_Toc514242784)

[GIT Commands Overview 8](#_Toc514242785)

[GIT initiation in project location 8](#_Toc514242786)

[Check file status 8](#_Toc514242787)

[Adding file into git 8](#_Toc514242788)

[Setting Global property in git 8](#_Toc514242789)

[Committing Changes in git 9](#_Toc514242790)

[Adding remote repo and pushing code into repo 9](#_Toc514242791)

[Verify it in github 9](#_Toc514242792)

[Git Branching and Merging 10](#_Toc514242793)

[Why Git Branch 10](#_Toc514242794)

[Working with git branch 10](#_Toc514242795)

[Create a project to explore branch 10](#_Toc514242796)

[Initiate Project via Git Bash 10](#_Toc514242797)

[Create a text file into this project 10](#_Toc514242798)

[Check newly added files status 11](#_Toc514242799)

[Adding file into git for tracking 11](#_Toc514242800)

[Commit file into branch 11](#_Toc514242801)

[Adding remote githup repo into git 11](#_Toc514242802)

[Push changes to master branch 12](#_Toc514242803)

[Github repo after push 12](#_Toc514242804)

[Explore Git Branch 12](#_Toc514242805)

[Creating branch 12](#_Toc514242806)

[Checkout Branch 13](#_Toc514242807)

[Add Modification into Branch 13](#_Toc514242808)

[Pushing new Brach 13](#_Toc514242809)

[Github repo after push 14](#_Toc514242810)

[GitHub Branches 14](#_Toc514242811)

[Git Merge 14](#_Toc514242812)

[Checkout master and Merge 14](#_Toc514242813)

[GitHub repo after Push - Master Branch 15](#_Toc514242814)

[GitHub repo after Push - New Branch 15](#_Toc514242815)

[Deleting Branch 16](#_Toc514242816)

[Deleting Branch from local 16](#_Toc514242817)

[Deleting Branch from GitHub repo 16](#_Toc514242818)

[GitHub Email Notification 16](#_Toc514242819)

[Settings in github repo 16](#_Toc514242820)

[Git Tag 17](#_Toc514242821)

[What is Git Tag 17](#_Toc514242822)

[Why should i create Tag 18](#_Toc514242823)

[When to create Tag 18](#_Toc514242824)

[How to create a tag 18](#_Toc514242825)

[Git tag 18](#_Toc514242826)

[How to create Annotated tag 19](#_Toc514242827)

[How to view/list tags 19](#_Toc514242828)

[Push tags to repo/remote 20](#_Toc514242829)

[Tags in repo before push 20](#_Toc514242830)

[Push tag to remote 20](#_Toc514242831)

[Remote repo after tag push 20](#_Toc514242832)

[Delete a tag 21](#_Toc514242833)

[Delete tag from local 21](#_Toc514242834)

[Delete tag from remote 22](#_Toc514242835)

[Remote repo after tag deletion 22](#_Toc514242836)

[Git checksum 22](#_Toc514242837)

[Create a branch from a tag 23](#_Toc514242838)

[How to create a tag from past commit 23](#_Toc514242839)

[Pushing commit point tag to remote 24](#_Toc514242840)

[Deleting renamed file 25](#_Toc514242841)

[Git Hooks 25](#_Toc514242842)

[What are Git Hooks 25](#_Toc514242843)

[Git Hooks categories 26](#_Toc514242844)

[Git Hook Folder 26](#_Toc514242845)

[How to Create your own hook 26](#_Toc514242846)

[Managing Hooks 27](#_Toc514242847)

[Explore pre-commit hook 27](#_Toc514242848)

[Create java and text files inside test project 27](#_Toc514242849)

[Commit with Java file 27](#_Toc514242850)

[Commit without Java file 28](#_Toc514242851)

[Trigger Build from Post-commit script 29](#_Toc514242852)

[Pre-requisites 29](#_Toc514242853)

[Get API token from jenkins Job 29](#_Toc514242854)

[Get auth Token from user configuration 29](#_Toc514242855)

[Create Post-commit script 29](#_Toc514242856)

[Verify post-commit scripts 30](#_Toc514242857)

## Daily Topics tracking

|  |  |
| --- | --- |
| **Date** | **Topics Covered** |
| 05/15/2018 | * Document formatting * git Commands in cmd line * git commands in git bash * git branching * git merging * git tagging   Pages ( 1 to 25) |
| 05/16/2018 | * What is hook and types * explore hook script * Pre-commit hook POC * Post-commit hook with api calls   Pages (26 to 30) |

## Git Installation

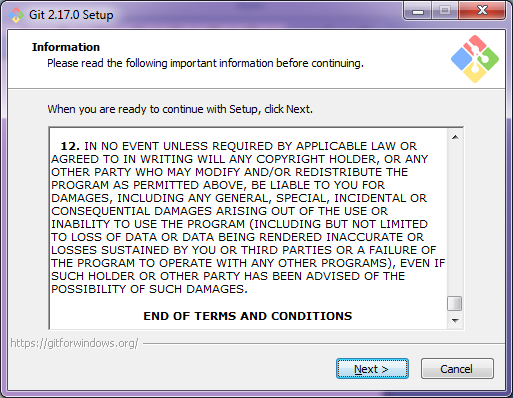
### Git download

GIT 2.17.0

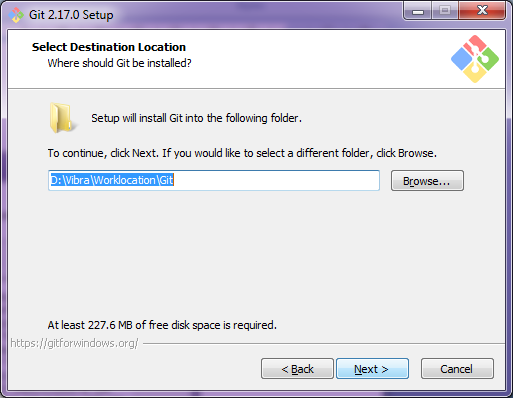
Go to https://git-scm.com/download/win

download windows version.

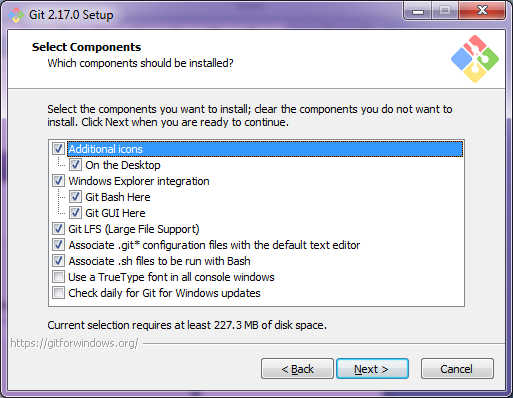
### Git Installation

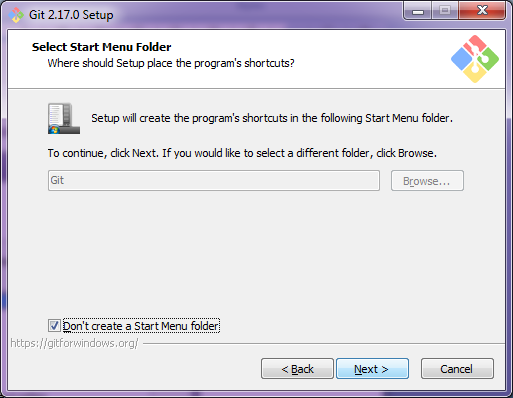


### Select Installation Path

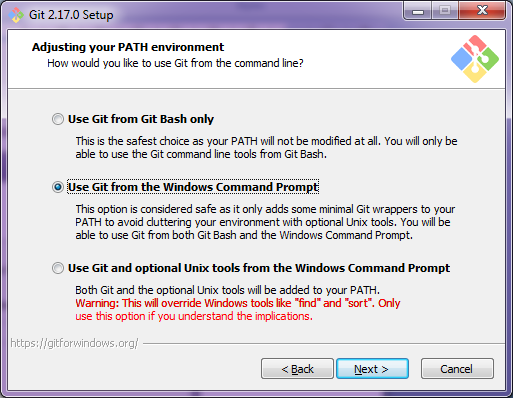


## Select Component for Installation

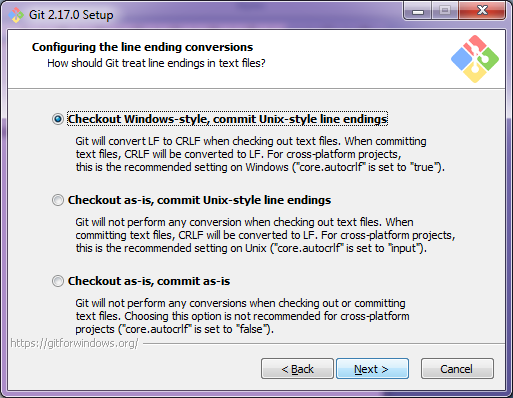


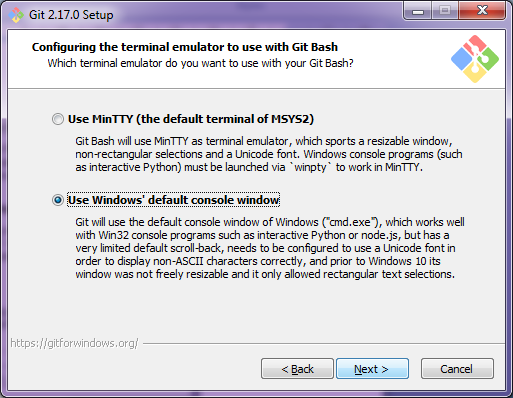


### Setting Path Environment

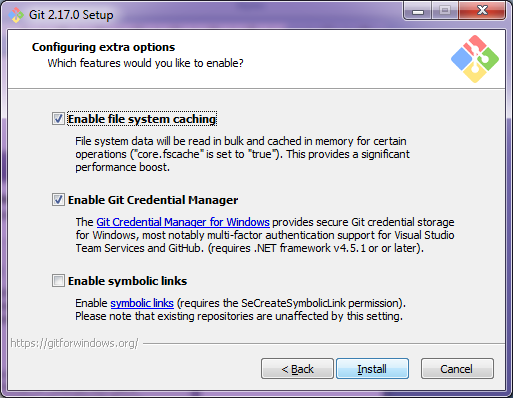


### Other Configuration set-up during installation



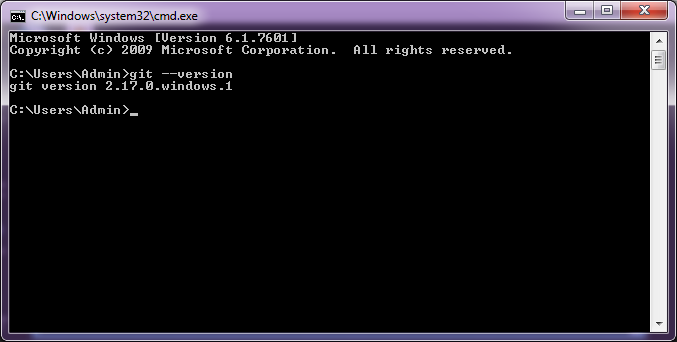


### Finish Installation



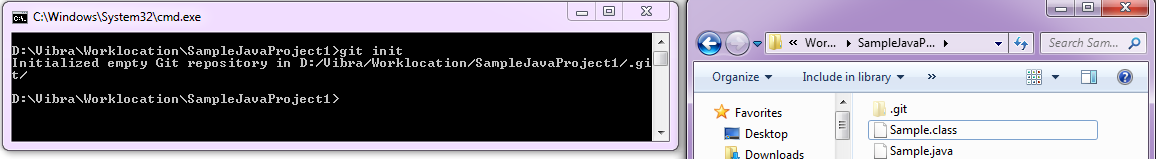
### Verify git installation

verify git version.

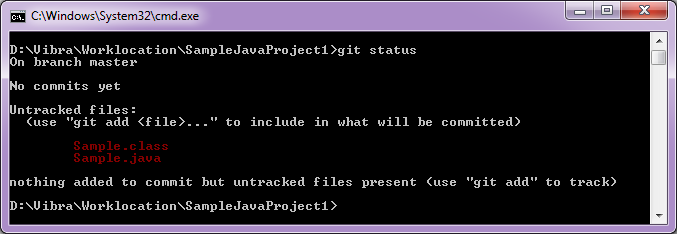


## GIT Commands Overview

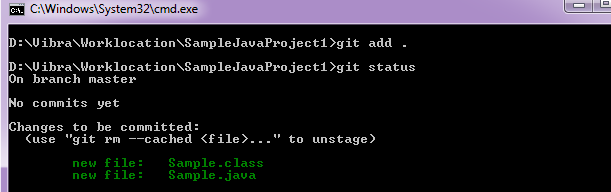
### GIT initiation in project location

  
execute git init inside your project location you can see .git created under your project folder this is the key for all your git transactions.

### Check file status

  
execute git status command inside your project directory. you can see files and its status. in above screen you can see 2 file available for commit and it is not added into git tracking.

### Adding file into git



Now you can add files into git by executing "git add ." this will add all new files.

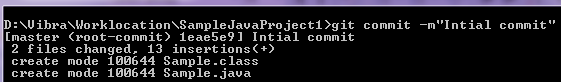
### Setting Global property in git

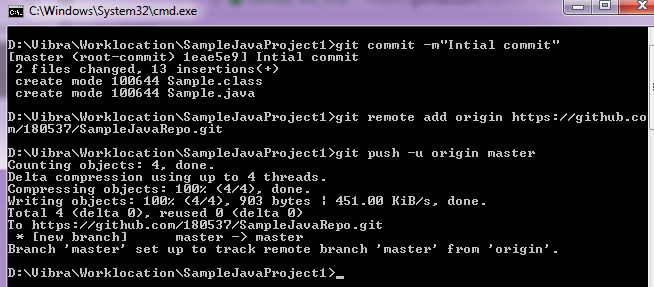
Before committing your change in git make sure your email and user name added/updated into git. you can achive this by executing below commands.

git config --global user.email "you@example.com"  
git config --global user.name "Your Name"



### Committing Changes in git

  
By executing git Commit you can you can commit files into git.

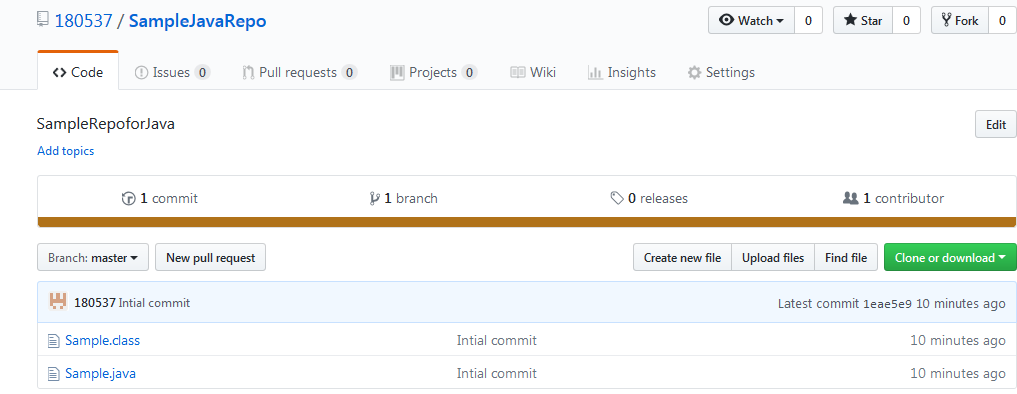
Adding remote repo and pushing code into repo

you can achieve this by executing "git add origin <repo location>" . you can note command/repo location during repo creation and you can use it here.

"git push -u origin master" will push your changes to master branch.

### Verify it in github

now you can check it in your github account, under newly created projects these files were added.



## Git Branching and Merging

### Why Git Branch

if team want to add a new feature or any maintenance work if is not recommended to all the code changes in main branch. instead branch will be created and all work will be carried-out in the branch

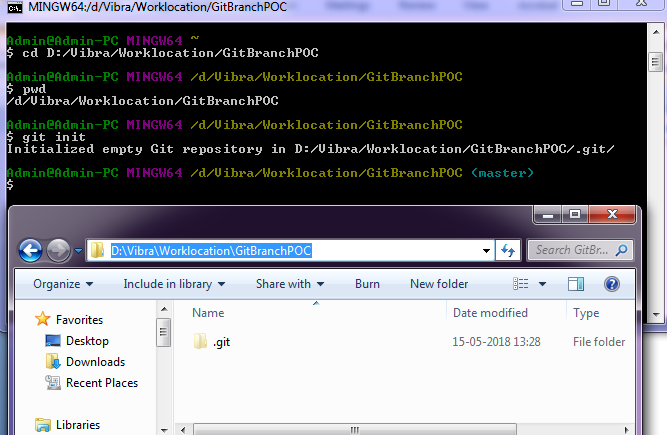
### Working with git branch

### Create a project to explore branch

Create a folder in your local to test this. in this case folder created in below path  
D:\Vibra\Worklocation\GitBranchPOC

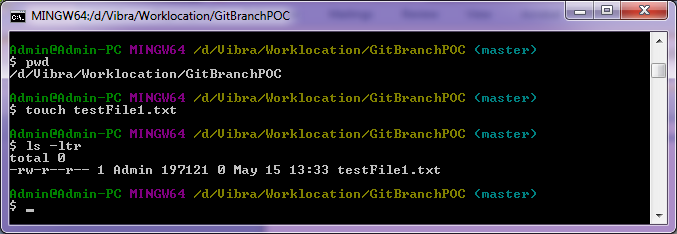
### Initiate Project via Git Bash

go to project location using git bash command line tool and execute git init command. This will create .git folder this is the base/key to track all our changes in git.

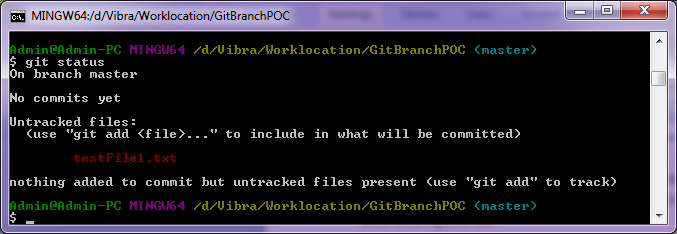


### Create a text file into this project

git bash works like linux prompt you can use linux commands. in below screen executed pwd commd to check the current folder and created one text file using touch command. ls -ltr executed after creating text file.

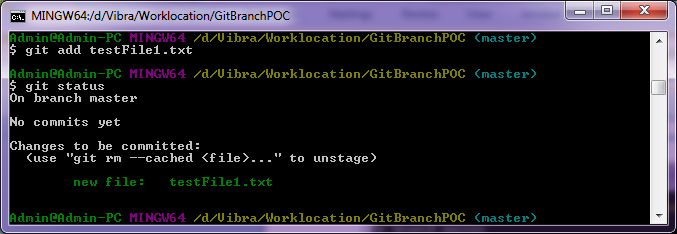


### Check newly added files status

execute git status command. This will show you newly added text file is available under un tracked list and this is ready for add into git.

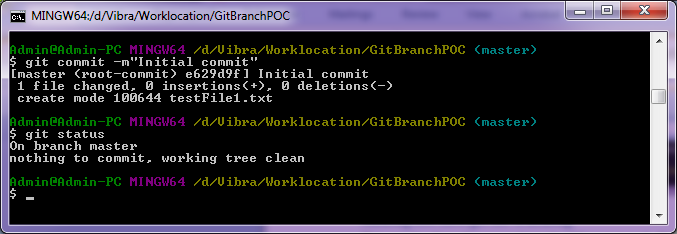
### Adding file into git for tracking

This can be done by executing git add command. file is ready to commit, this can be verified by executing git status command after git add.



### Commit file into branch

execute git commit with message -m can be used for message. after commit verify the status. No files available, and system is good with previous commit.



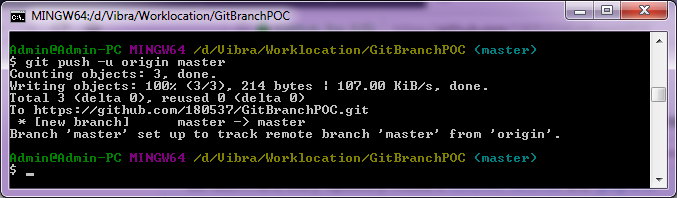
### Adding remote githup repo into git

Adding remote repo to git create a repo in github and execute below command in git

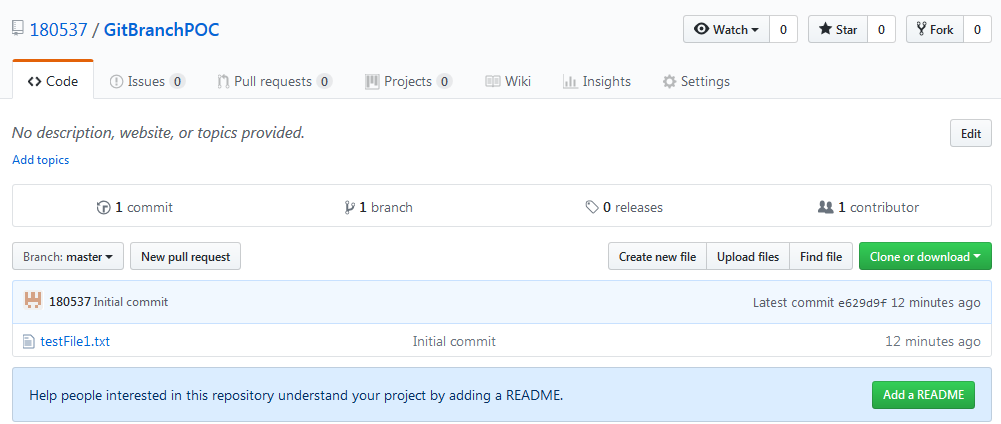
git remote add origin https://github.com/180537/GitBranchPOC.git



### Push changes to master branch



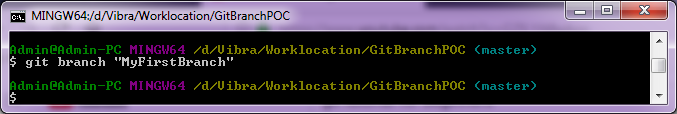
### Github repo after push



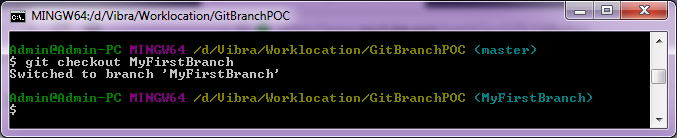
## Explore Git Branch

### Creating branch

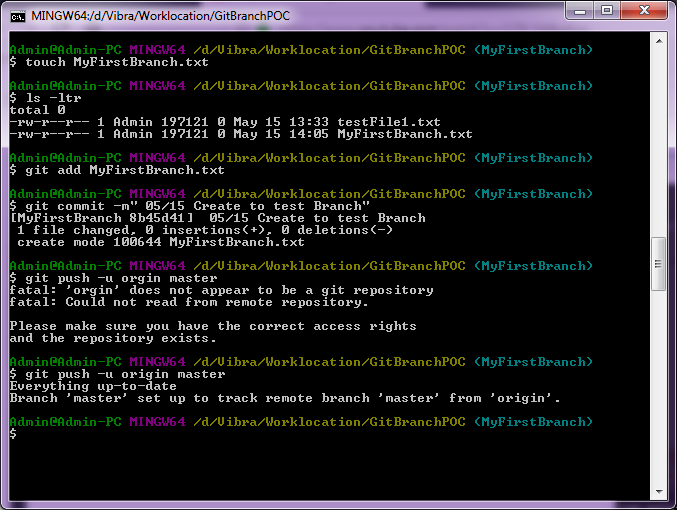
This can be done by executing git branch "BranchName".



### Checkout Branch

this need to be done to do changes in newly created branch. git checkout "branchname" will create a branch from master. user can do their change in it.  


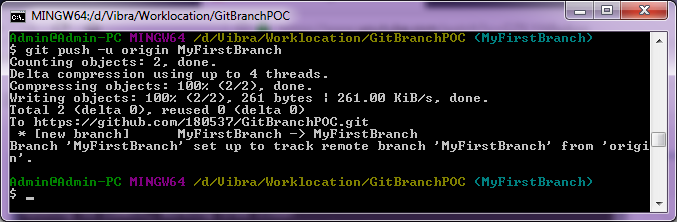
### Add Modification into Branch



new file created in our new branch by executing above commands in sequence. you can also see push to origin shows as master is update to date and new branch changes not included. up to this point newly created branch will not be available in github.

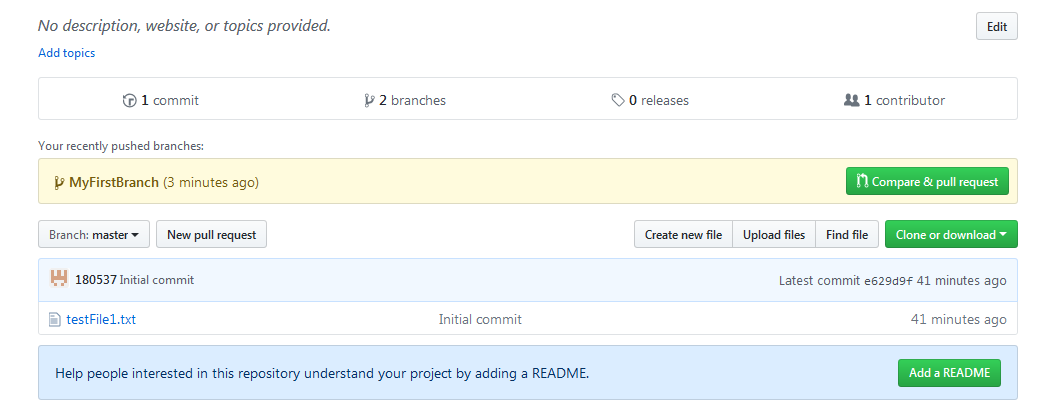
### Pushing new Brach

This can be done by executing git push -u orgin "branch name"

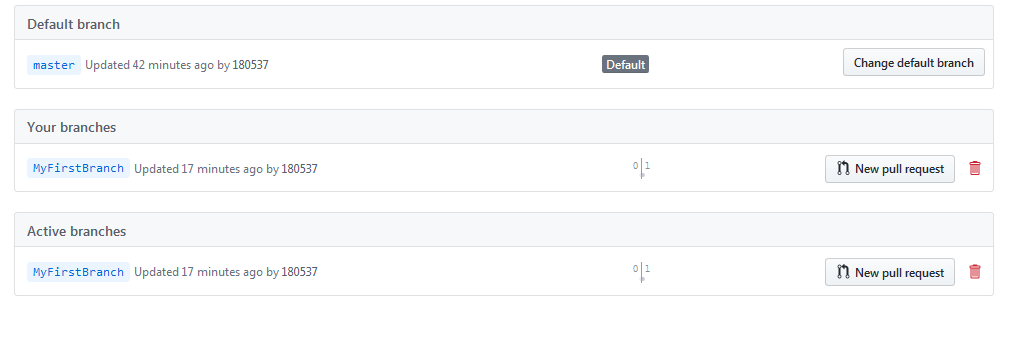


### Github repo after push

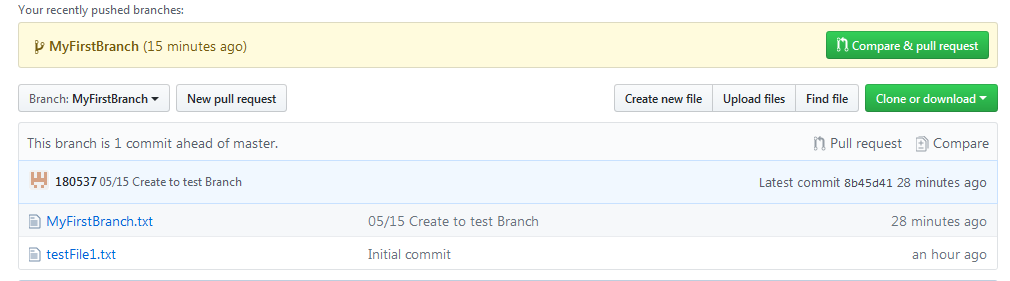
Branches count increased to 2



### GitHub Branches



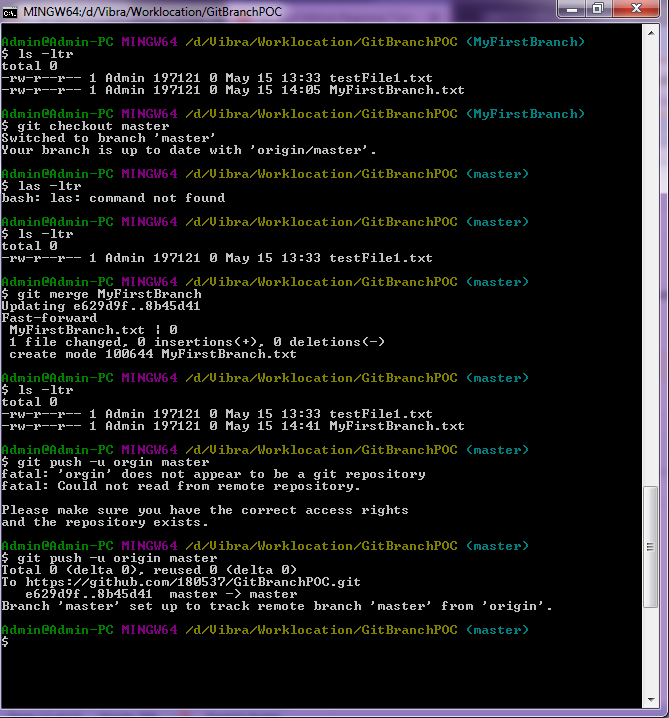
Newly created branch will have latest file however this file not available in master until merge this branch with Master.



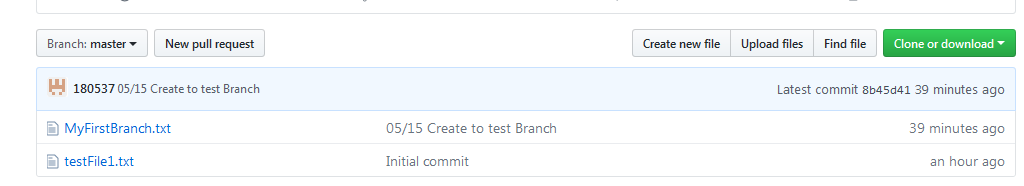
## Git Merge

## Checkout master and Merge

to merge checkout master and do ls -ltr, this will list files in master. and do git merge "branch name". This command will add change into master. then push those changes to master.

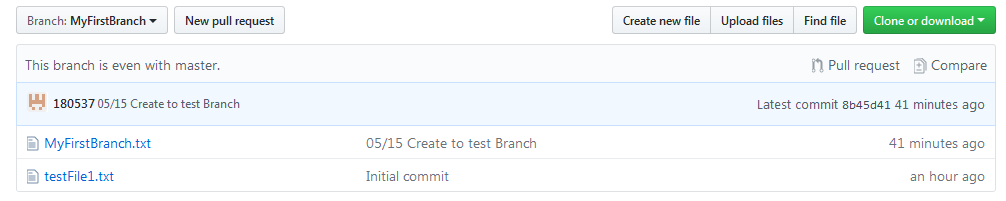


### GitHub repo after Push - Master Branch



### GitHub repo after Push - New Branch

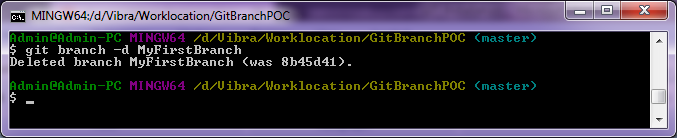
Brach is even with master text at the top denotes both branches are identical.



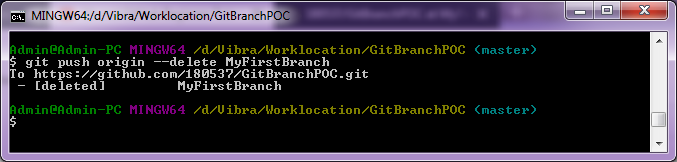
## Deleting Branch

### Deleting Branch from local

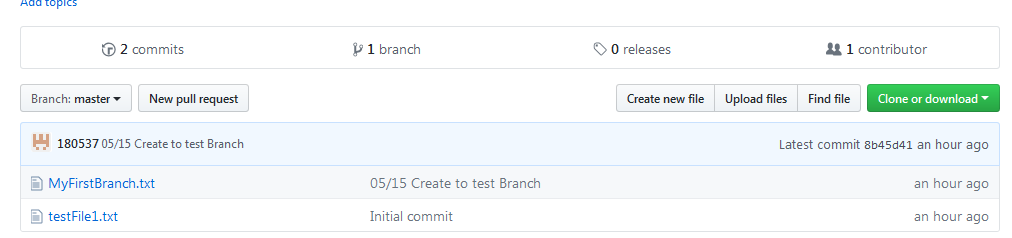
by executing git branch -d "branch name" will delete branch from local. however after deletion, this branch still exist in GitHub repo.



### Deleting Branch from GitHub repo



Branches count decreased to 1

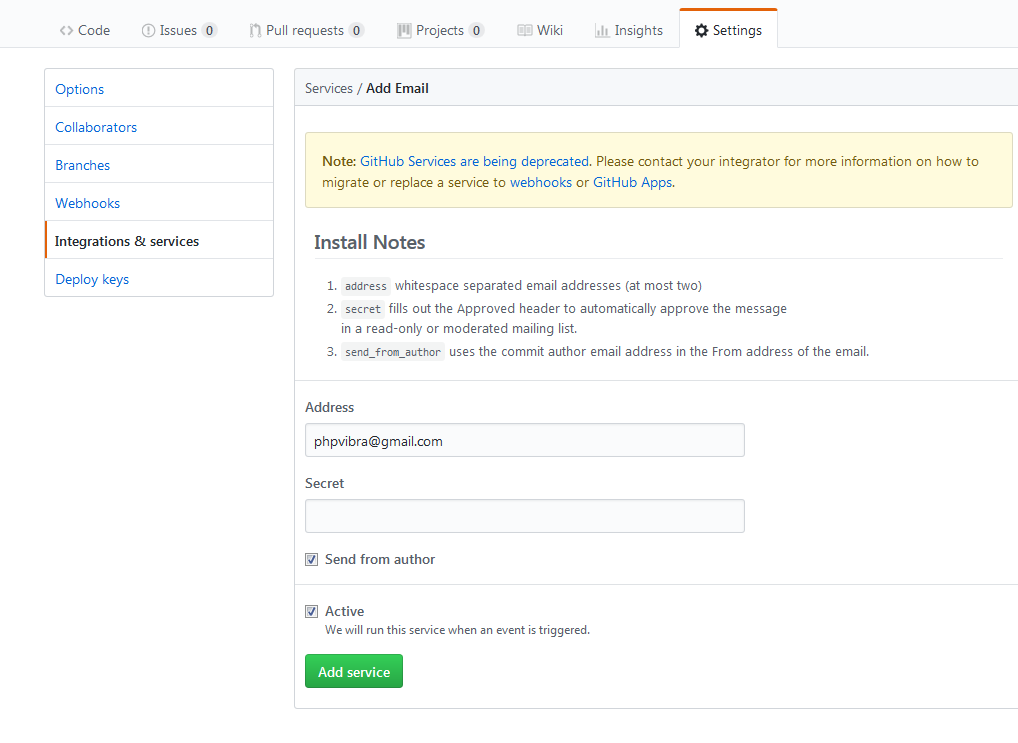


## GitHub Email Notification

### Settings in github repo

Select a repo to add email notification and follow below navigation

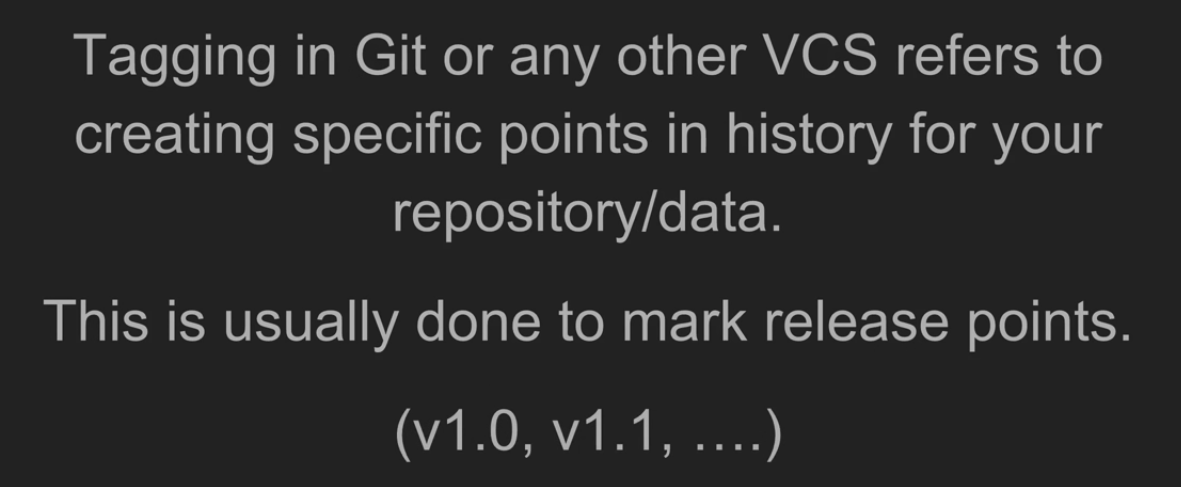
Settings >> Integration & service >> click ADD service button >> type email and select.



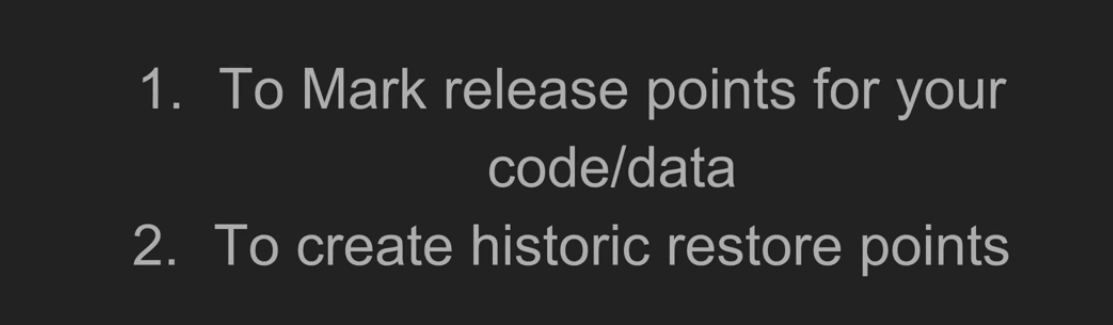
Enter details as mention below and click add services. This will add a service.

## Git Tag

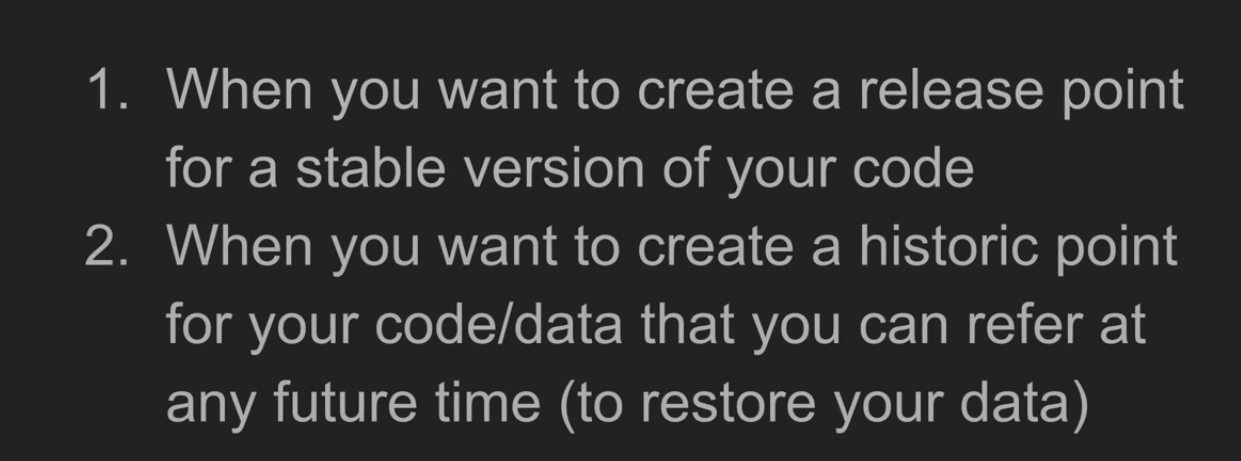
### What is Git Tag



### Why should i create Tag

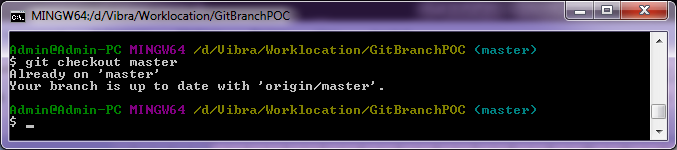


### When to create Tag



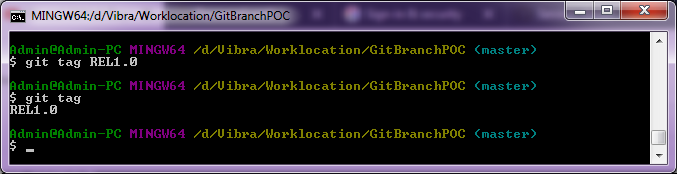
## How to create a tag

Checkout a branch in which you want to add tag.



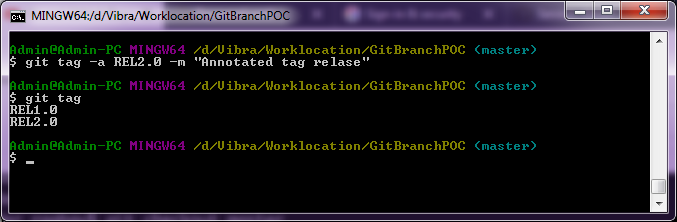
### Git tag

execute command git tag "tag name"



### How to create Annotated tag

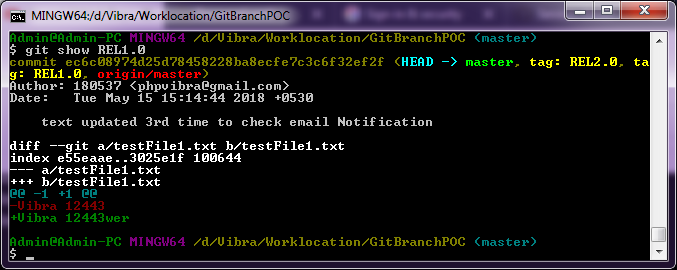
this can be done by executing git tag with - a



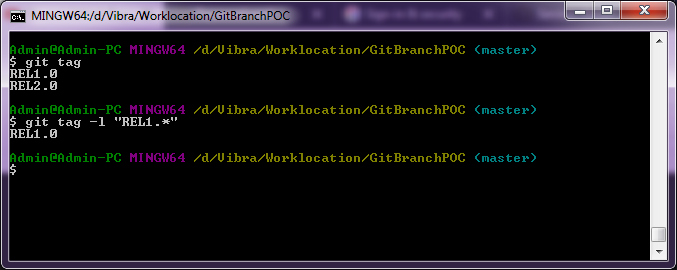
### How to view/list tags

Git tag will list available tags in a branch. refer above screen.

Git show <tagname> will show tag details



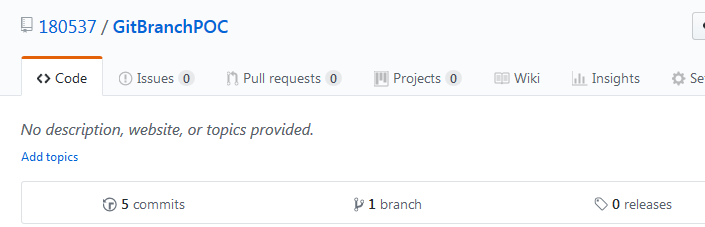
#### Git tag with wildcard git tag -l "REL1."



## Push tags to repo/remote

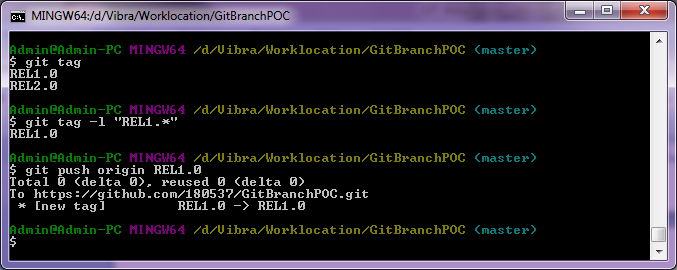
### Tags in repo before push

there are 0 released in remote repo.



### Push tag to remote

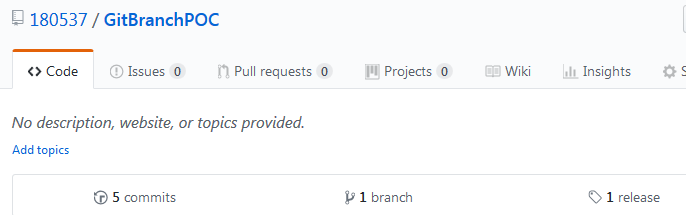
git push origin <tagname>



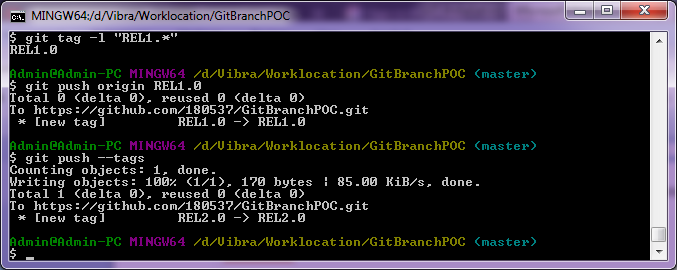
### Remote repo after tag push

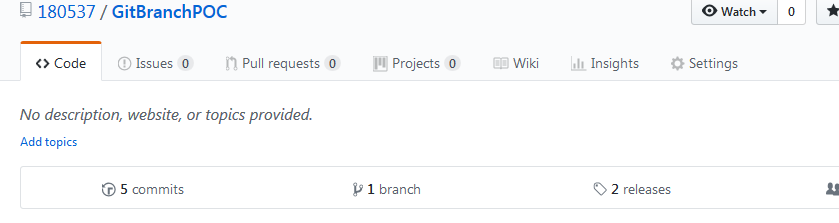
Release count increased to one.

git push origin tagname or git push [origin] --tags to push all the tags to remote



#### Git push --tag

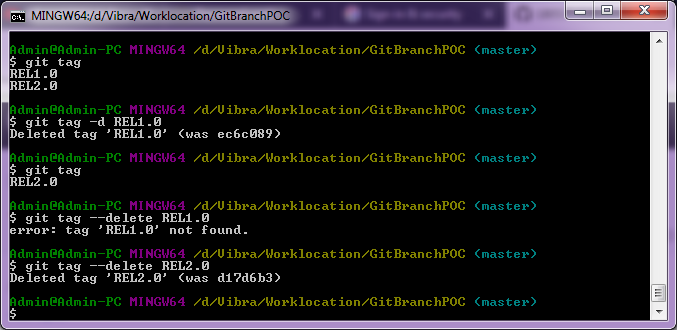




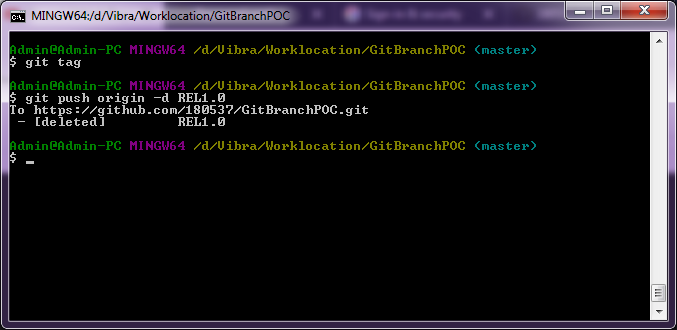
## Delete a tag

### Delete tag from local

git tag -d tagname or git tag --delete tagname or git tag -d tag1 tag2

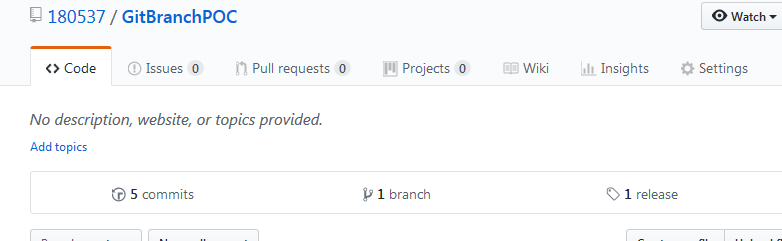


### Delete tag from remote

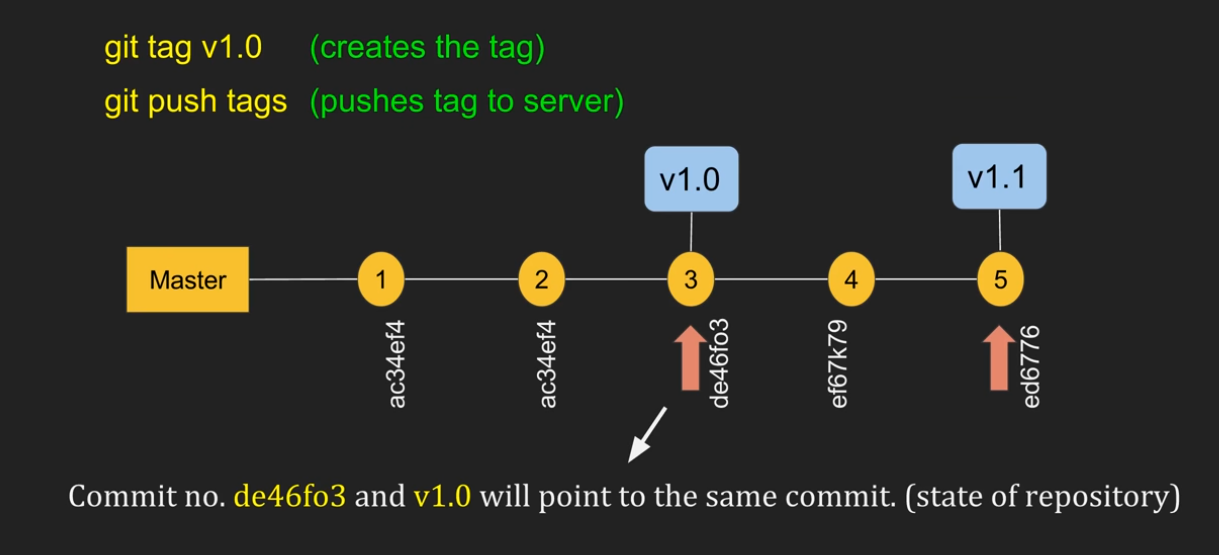


### Remote repo after tag deletion

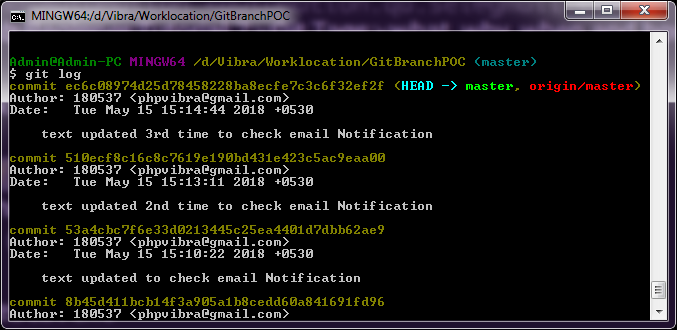
Release count decreased to one.



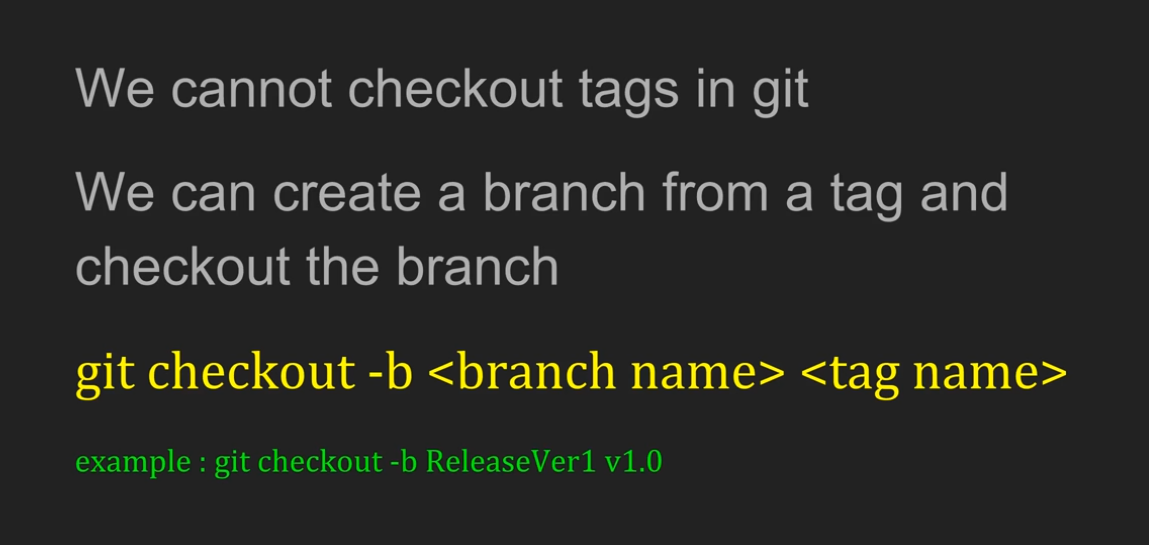
### Git checksum



execute git log to see every commit. in every commit you see 40 digit number. this is called a check sum and it is unique to a commit

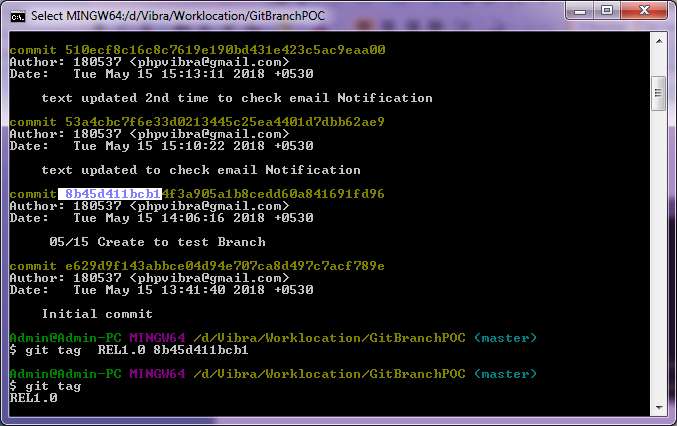


## Create a branch from a tag

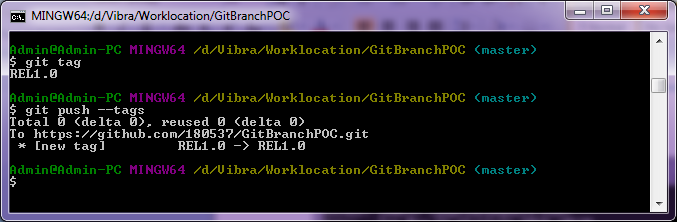


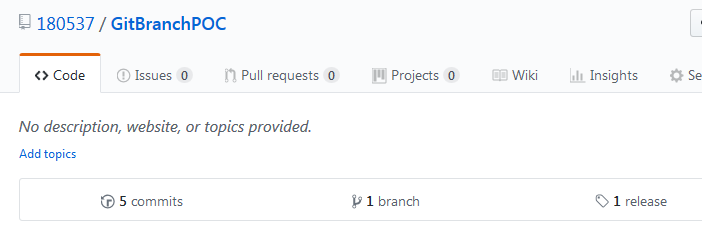
## How to create a tag from past commit

do git log and select any one of the checksum and create tag. refer below screen selected few from 40 digit checksum and created tag from that checksum.

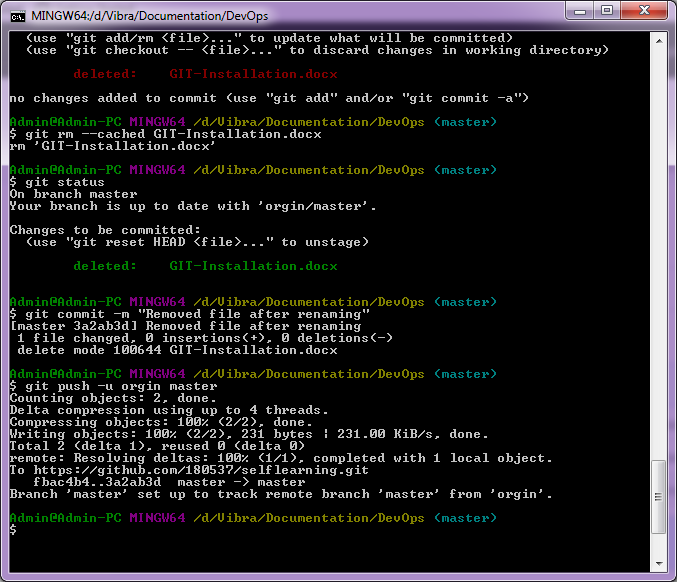


### Pushing commit point tag to remote



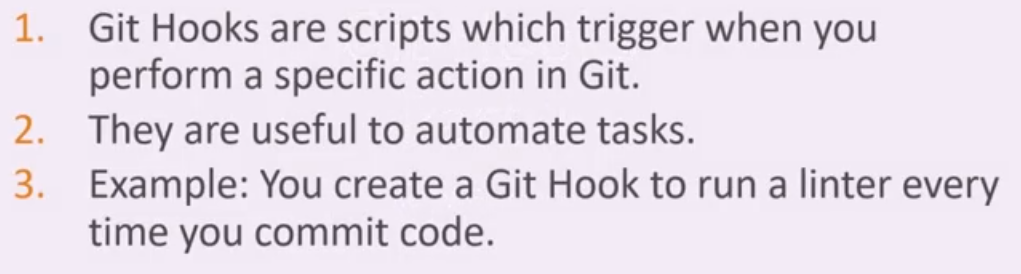


## Deleting renamed file

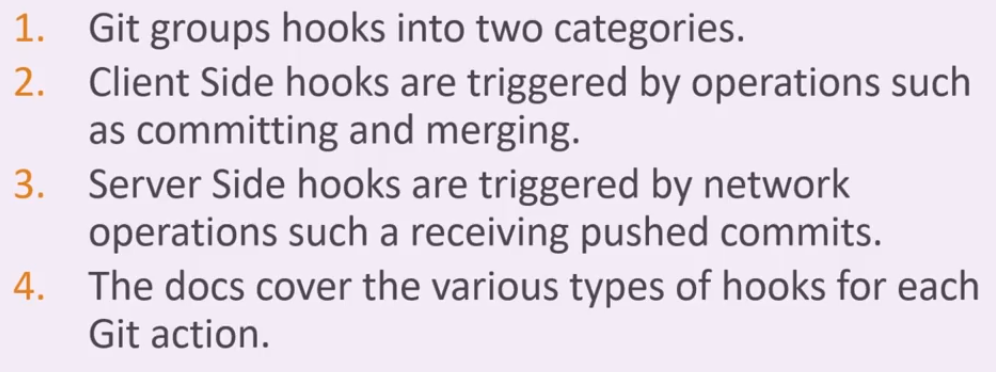


## Git Hooks

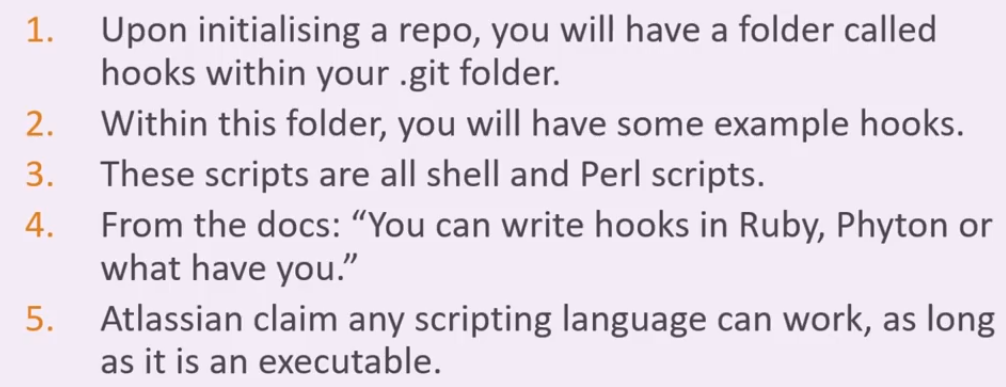
### What are Git Hooks



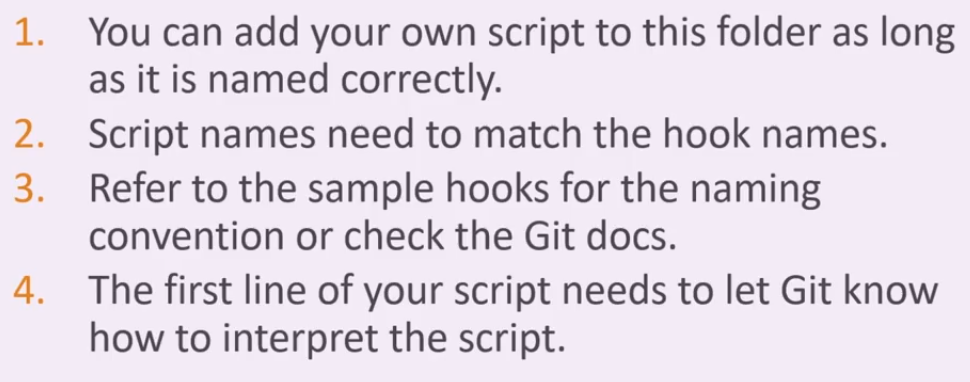
### Git Hooks categories



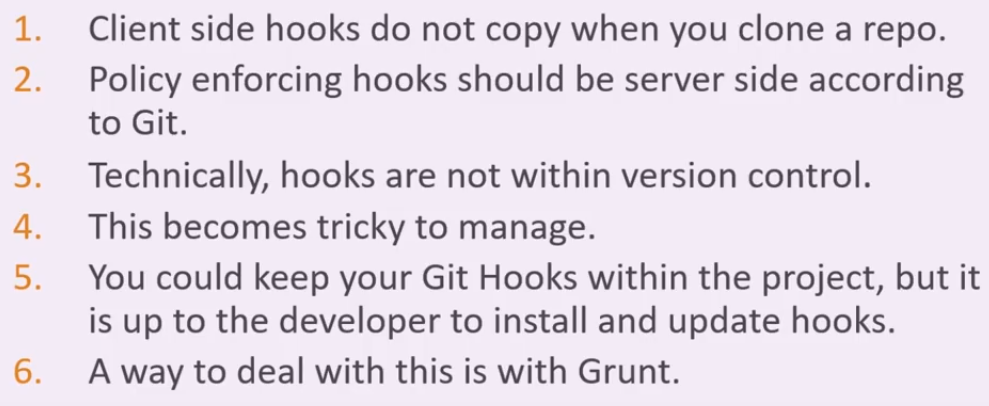
### Git Hook Folder



### How to Create your own hook



### Managing Hooks

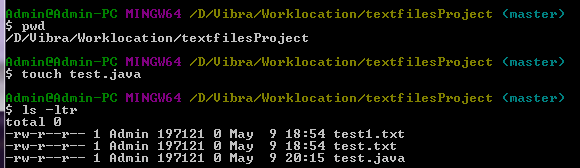


## Explore pre-commit hook

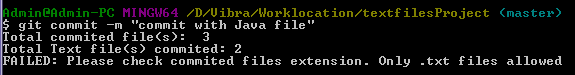
create pre-commit file inside .git/hooks folder. created one pre-commit hook to check user committing only text file. if any other files committed commit will fail with error message. below script is a sample you can change the script based on you experience in shell/bash script.

  
This script will automatically triggered when user try to commit.

### Create java and text files inside test project

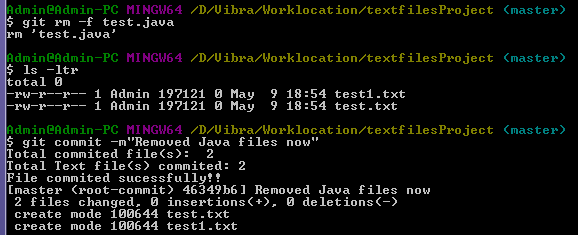
 you can see 2 text files and one java file created in textfilesProject.

### Commit with Java file

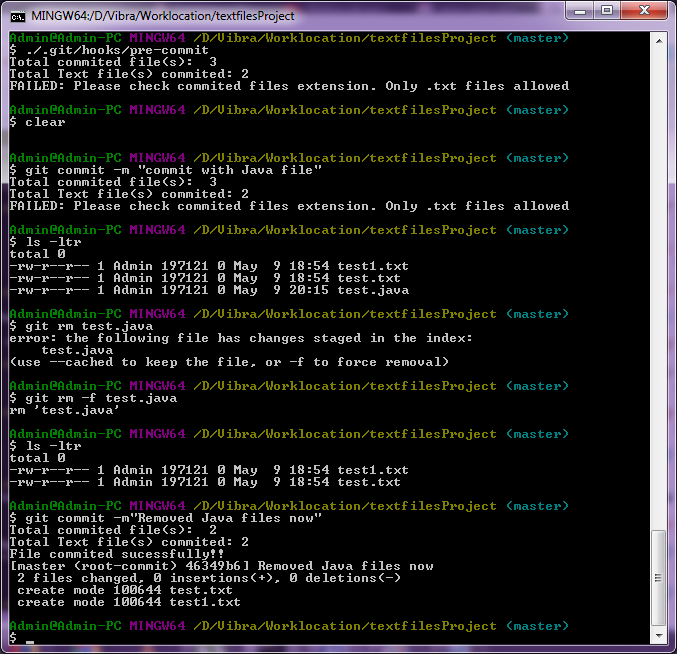


as expected this commit failed and got error custom error message and count of total files committed and txt files count.

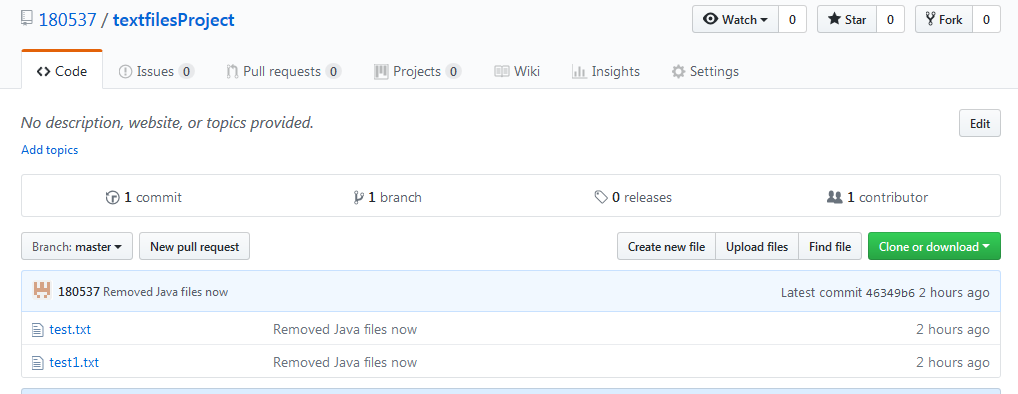
### Commit without Java file



Removed Java file and committed, as expected pre-hook script allowed this commit and got total count and text file count and custom message.



After git push these files added in to github.

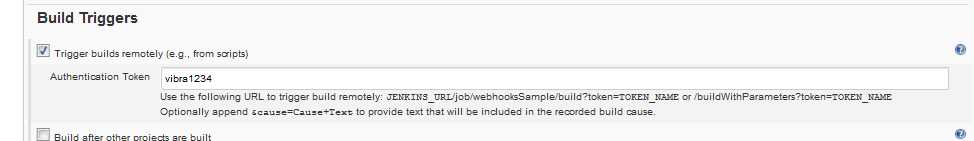


## Trigger Build from Post-commit script

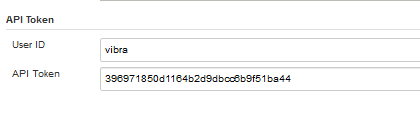
### Pre-requisites

### Get API token from jenkins Job

get API token from build job by selecting Trigger builds remotely. and form the Jenkins URL. For first time users you can set you own API token and Save.



### Get auth Token from user configuration



### Create Post-commit script

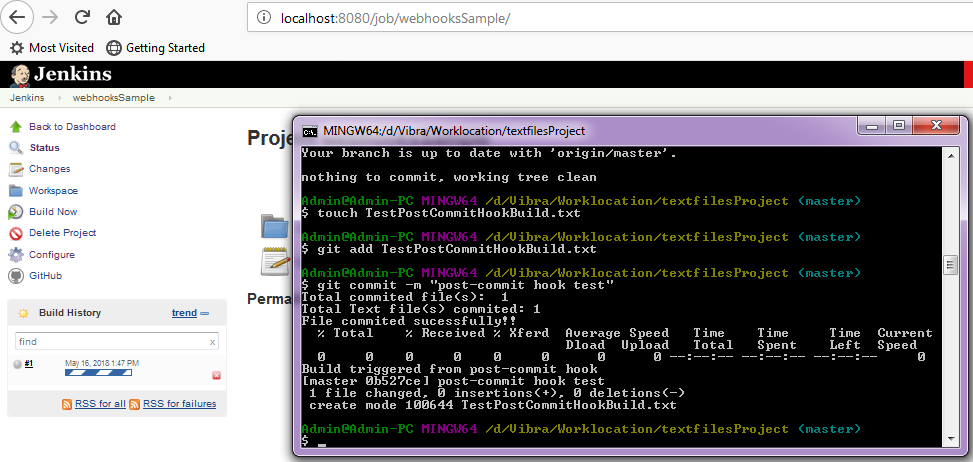
go to .got folder and create post-commit file and add your script to trigger a build. in this POC, build trigger initiated via curl command with auth token and API token mention above. you can use below attached post-commit hook for testing.

curl --user vibra:396971850d1164b2d9dbcc6b9f51ba44 http://localhost:8080/job/web

hooksSample/build?token=vibra1234



### Verify post-commit scripts

  
In above screen, you can see post build scripts triggered build when commit happened. Also you can notice message from "Build triggered from post-commit hook" which we added in post-commit script.