251. Introduction to GIT

GIT - it's a free and open source distributed version control system.

GIT plays a crucial role in framework. when I say framework, this has nothing to do inside the framework but when you have developed a robust project it could be project or folder or framework, how do you distribute this across your peers?

so in real time when you work on a project there could be a number of people working on it,

So how do you track the project? Like if one guy makes an update how will other person get notified and how do you merge the both changes into one common repository.

So without Git and just like Git there is one more tool called SVN. Both the same thing, distributed version control systems. Most the people using GIT only.

So you cannot imagine any project, any framework, without Git or SVN

Google – git download – download it and run the file – click n next next and install and do setup.

EX:

Person x - writes some code and create some testcases – EST Timezone

Person Y – IST timezone

Not only Y multiple people

So first, X started his work and have done some progress and he left.

So then IST time zone begins and now Y will come to office and he has to pick from where X have completed his work.

So if you want to pick the test where he had left he has to leave it one central repository

from where this guy can download.

GitHub is a central place where you can post your code so that, this Y comes to office next day and take the latest code from the GitHub.

252. Importance of Github and its uses

Login to GitHub here where you will post your code and you will access from your local machine using Git commands.

Ex:

assume that Y have written one to five lines of code which belongs to first component.

So they shared work like that. So X has to develop second component and which comes from fifth to tenth line of code. When they share work like this, in his code, he will simply write code for second component. In Y machine, he will have code only for first component

in test case A. So when he pushes the code, automatically the second component will be stored. But he have done some coding, But when he pushed the code, in general, he will see only the second component but his first component will be overridden, if he take the code.

But with GitHub, it have a smart intelligence to understand that what is that code extra you are getting while you push the code and what is the code you're already having and it will merge this two code and neatly forms a one test case called A with one to 10 lines of code,

starting with first component and second component, This completely merges your code.

merge will be done automatically by Git, but sometimes there is a chance that even Git confuses, that every time whatever code you send, it will merge. So 90% of times, Git will take care of merging, but sometimes it will throw an error, saying there is a merge conflict

and it will ask us to fix that conflict and then push the code.

253. Creating Git config and repositories

Login to github create one new repo(repo name : GitDemo ) – click on create repo.

so now Git demo repo is ready with empty code and now we have to push the code

from our local machine which we are working on to this Git demo so that everyone can use it.

So first you need to tell Git who you are. So before you do any operation, Git has to know

from which user I'm doing all these communication to GitHub and what is his email ID.

you need to give before you start working with Git in your machine.

Let's do what Git actually provided initially for talking with GitHub.

[Basic Git commands | Bitbucket Data Center and Server 8.17 | Atlassian Documentation](https://confluence.atlassian.com/bitbucketserver/basic-git-commands-776639767.html)

Open command prompt

Config is a command basically used to set name and email.

And obviously, you have to start every command with Git.

I am setting global because I can access this anywhere from my system.

|  |  |  |
| --- | --- | --- |
| [**Tell Git who you are**](https://www.atlassian.com/git/tutorials/setting-up-a-repository/git-config) | Configure the author name and email address to be used with your commits.  Note that Git [strips some characters](http://stackoverflow.com/questions/26159274/is-it-possible-to-have-a-trailing-period-in-user-name-in-git/26219423#26219423) (for example trailing periods) from user.name. | git config --global user.name "Sam Smith"  git config --global user.email sam@example.com |

254. How to push code to remote repository

 I will create one folder in my local system name ( GitStuff ) .Inside this folder add few files for demo purpose

255. Understanding Staging and commit in git

So I'll go to my command prompt and I'll go to the folder where cd Gitstuff.

Git init is a command. Let's give this and then it will initialize a file into your folder and you'll see this Users/rahul/Gitstuff/.git. You see this .git file, that is the file created in this folder.

and now this is all set to push to the GitHub.

There are two terminologies in Git which is stash and commit.

without commit you cannot actually push the GitHub.

what is Stash?

First, when you actually develop something this commit accepts the files from Stash.

This is like first level of commit, and this is final level of commit, to say it technically,

So generally for commit one we say it as Stash. Like now when you want to commit,

commit looks only for the files which are in the stash and then it commits.

You know this order starts like this. GitHub only takes the code, which is committed.

Commit, commits the code, which is present in Stash only.

So now first if you want to push any code to GitHub, first you need to add your code to the stash and then you can commit the code from the stash and place it here.

This is commit level two, and then you can push it to that directly to GitHub. So this is the order, it starts from normal to stash to commit to GitHub, you have to follow this path

to make your code updating GitHub, finally.

Git add star. So add is a command which says that add your files to stash, and star represents add everything.

If we use git add \* entire code or all the files present in the git stuff tolder is moved in to stash.

git status and to see what are the files added to the stash.

Git commit -m “first commit” – now  we are all set to push our code now to GitHub repository.

256. Add remote repository and push the committed code

Now you need to push, but before pushing, you have to tell to your local Git that where is an address, like where exactly I need to push,

when you create repository, by default, you will be on master branch

251 t0 256 entire code

First download git then create git hub account and create one new repo then

open command prompt

git config --global user.name "hari"

git config --global user.email "harisankarayyapureddy@gmail.com"

cd Gitstuff // create one folder(Gitstuff) in your local system and place few of the files

git init

git add \* // \* means add all files present in the folder ( this command we call it as stagging )

git commit -m "commit message"

git remote add origin <https://github.com/harisankar722/Git-Demo.git> ( in real time this patch will give team)

git push origin master // master is by default branch

git status

257. End to end working example on Git commands -1

 project is ready in our repository so our next step is to clone this project in another mission.

X has write the code and push in to the repo that is gitstuff Y candidate came want to take this code from GitHub and continue his scripting, how to do that ?

if you want to clone that code go to command prompt

Clone is something used to extract the code present in the repository for the first time,

Why I am saying first time ? Later point of timesif you want to take the code

you will use pull command, Pull only the updates which happened after you cloned for the first time. When you cloned for first time there is nothing, It will give you everything.

But thereafter you have to use pull. So that pull checks what is new from your already present code in your system. And that new code only will be downloaded instead of downloading again whole code, Then we use pull.

Cd ..

Git clone link ( link in the github project )

One you run these two commands x person code is come to in your machine

Open eclipse -> file->import->General->Existing project in to workspace->next->Browse the project->select git demo project and k->finish

Open two workbench means two eclipse only one person is in one workbench

Same import steps what we use same thing repeat it in eclipse and here you browse gitstuffx.

we know that GitX is the one who have written fresh code and uploaded, and in GitDemo he downloaded. So because he has to continue his work. I will open the project and I'll make some changes.

258. End to end working example on Git commands -2

I will go back to my Git X project now and I will do git pull because I want to take the latest code.

if you want to take entire repository for the first time, you can use Git clone.

But here you need not get entire repository, you just need to get latest changes what happened in the framework. Because we already have the code in the Git X.

Cd ..

Cd gitstuffX

Git pull origin master

257 & 258 explaining is – GitstuffX person moved the code in to GitHub after that GitDemo(y person) came and take the code what X person uploaded Y person take the code and do some changes and uploaded in the Github then again X person take the latest code in the github repo.

259. Importance of Branching in GIT

 GITX and GITDemo, two people working from different time zones and collaborating each other with the help of GitHub central repository. - these guys are concentrating on test cases development.

So management decided to make some modifications on framework itself.

To make it more optimizable or to introduce some new futures. So they want to implement

and enhance this framework to the next level.

architects have to now focus on the project structure, like introducing a new futures

like test and (indistinct)or they want to drive the whole project from page object.

So they want to make many changes. But when these guys started touching

the actual core framework files, obviously these scripts will start fail,

Now instead of data driven framework if these guys are trying to implement page object of,

they will change some configuration files in the framework,

If they change test cases will fail. So they need at least one month to come up with a new framework, but in that one month these test cases will not run because of these guys' changes.

Because they're making changes component by component. But these test cases are necessarily

to run on everyday basis. They have to run on an everyday basis just because of some deliverable commitments. In that situation we will take a branch.

Master Branch -. sub branch will actually have the same code of what you have taken from original branch. Master is our root branch.

Master branch will continue to serve a regression test cases on daily basis.

At the same time, these guys have taken one more branch called develop branch.

You could name it with anyone. But in standard people use the develop if it is a second branch after master. So develop branch, they're developing the stuff.

In the same way, if another architect have to catch up with this work, he will actually pull the develop branch instead of master branch. Instead of master, these guys will work and develop

and they'll complete. So after one month, once they realize that this branch is fully ready

and executes all our test cases, they simply merge the develop branch into the actual master branch. After thorough testing, making sure this branch is not breaking, they will merge entire branch into the master branch.

Git checkout -b develop ( develop is the new branch name )

-b stands for creating new branch.

Git branch – if you want to know on which branch your in

it'll create one more branch It copies actually as one more code repository but it named it as the develop branch so that you can switch to the develop branch and you can work on entire that specific branch. And once it is done, you can merge that into your actual master branch

so that you are not disturbing the regular deliverables by taking cutting separate branch here.

So automatically, when you switch to the develop branch and when you go to your GITDemo,

your Eclipse also will shift. Whatever changes you make you will not disturb the original branch.

if you want to switch to a new branch which is already created - git checkout branchname

master is only taking care of regression test cases and this taking care of framework.

Now both are ready and now we need to merge this develop branch into master

and make this as one powerful framework with all the framework changes done by architect.

So how to do that?

switch to master branch first git checkout master. And make sure you have the latest code

you can do that git pull origin master.That just making sure that we are having the latest code

which is presenting GitHub repository.

Now if you want to merge develop branch once it is completely ready into your regular deliverables master branch you can simply do that with the git merge,

which branch you need to merge? Branch name ( git merge branchname )

260. How to resolve Merge conflicts with GIT

Create new text file in the eclipse project - alphabet a,b,c,d,e,f - save it in one of the package ( names.txt)

Then git add copy the path for the text file enter

Y person pull this code and he add some new numbers 1,2,3,4,5,6 after again x person removed e, f then Y person things my work is done I will merge the code Y person don’t know X person removed e,f once Y person merge then we will get the merge conflict msg.

I'm creating one new branch. So when I create a new branch, the whole code, whatever you have

from a master will create a copy and store that in branch.