Yesterday's session : Jenkins workflow in Realtime

-----------------------------------------------------

-> As a developer, we will send an email to git admin team to create repository for our project. All developers will keep project source code in git repository.

-> After Git repository got created, we will send an email to DevOps team to create jenkins jobs for our project.

-> In Realtime, project contains multiple environments like below

a) local env

b) dev env

c) qa env (Quality Assurance)

d) uat env (user acceptance testing)

e) pilot env (pre production)

f) prod env (Live env)

-> Our system is called as local env where we will do development and we will perform Unit Testing of our code.

-> After unit testing is completed our code should be moved to Git repository.

Note: All the developers code will be integrated in git repository.

-> All the developers code will be deployed to DEV env

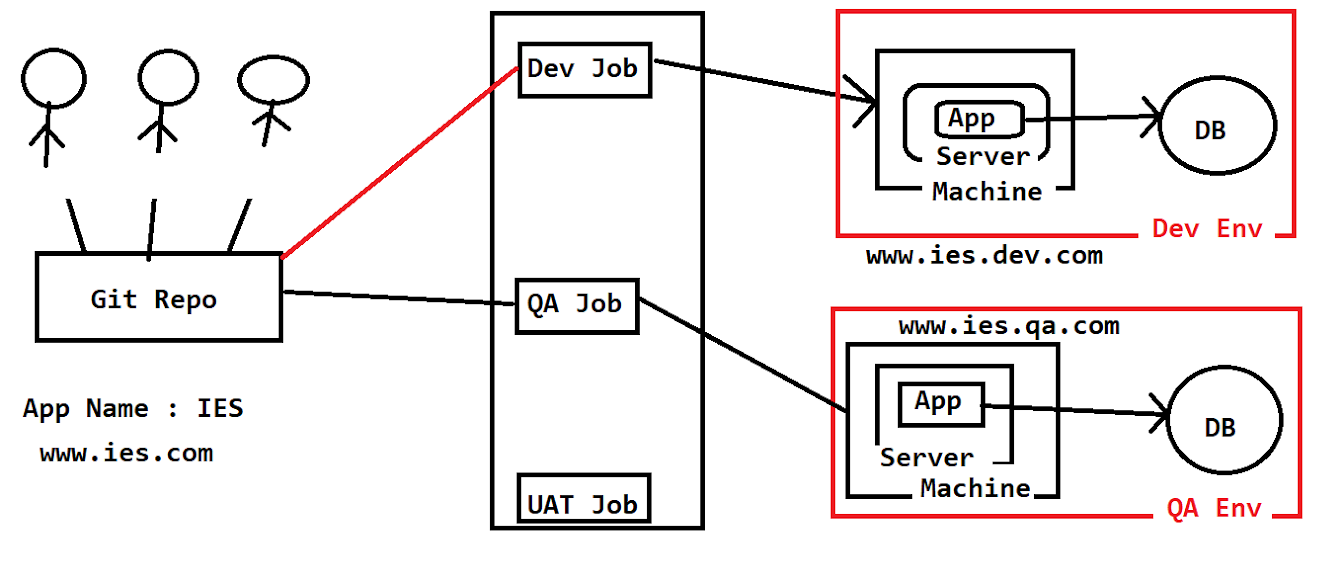
-> All the developers are responsible to test their code in DEV env. That is called as Developer Integration Testing (DIT).

-> After DIT is completed, our application will be deployed into QA env so that testers will test our application.

-> After QA testing is completed, application will be deployed into UAT env so that client will test our application

-> After UAT testing is completed, application will be deployed into Pilot env. (Pilot is called as Pre-Prod env)

-> After Pilot testing is completed, applicatio will be deployed into Prod env. Production env is called as Live environment.



Yesterday's session : Application Environments

----------------------------------------------

-> In realtime , for every application multiple environments will be available like below

1) Local

2) DEV

3) QA

4) UAT

5) PILOT

6) PRODUCTION

-> To deploy our application to these environments we will use Jenkins jobs

-> Application Environments setup will be done by Admin

-> Jenkins jobs will be created by DevOps team.

* For every env dedicated Jenkins jobs are available
* Jenkins jobs will be created by develops team

Today’s session : Version control tools

* In real time project, multiple developers available for development activity
* The developers maybe working from different different location for the same project.
* All developers code should be stored in one place
* We need to monitored access for source code to know who modify the code, deleted file ..etc

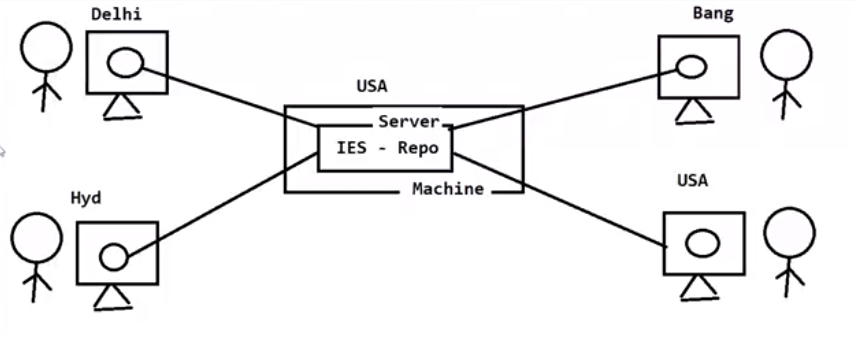
To have code integration and monitored access we will se version control software

As you can see below all developer are working from the different location and keep their code at one place that place is called

Repository and code integration

Repository: It contain source code of the project

Version control: It is a server who maintain all changes, modification of the code in detailed information



There are several version control software are available

1. SVN (sub version) 🡪 out dated
2. GIT HUB
3. Bit Bucket

* Only one software used for version control in the project, SVN is almost outed

# GIT Hub :

* It provides software project hosting and version control management using GIT
* GIT HUB : it provide web application to create a account
* If we create account in GIT HUB and it will provide some free space in cloud to create repo for source code
* It is also provides commercial plan also for source code management

Yesterday’s session: GIT HUB Introduction:

Today’s class : GIT Operations

* What is GIT HUB : it is a cloud platform where we can store our project source code in the repository
* GIT HUB uses GIT as a version control software
* In GIT HUB we will create repo for storing source code
* All developer will install git client software to interact with GIT HUB repository
* Create a repository

1. Login to git hub --. Click on new 🡪 give the name for repo and set is it public or private repository ?
2. You will get REPO URL : <https://github.com/pradeep2644/JavaRealTimeRepo.git>

* Git commands:

With help of git commands, we are going to use to perform operations in git repo

1. git init : It is used to initialize our folder as git folder
2. git status: it is used to check untracked files
3. git add: it is used to add files in staging area to make files eligible for commit
4. git add remote : to add remote rep URL
5. git commit -: to commit files to local repo
6. git push : to push files from local to central repo

Git Bash commands :

1. we’re having several git commands to perform operation with git repository

Note: we have created git repository in github.com

1. get repo URL : <https://github.com/pradeep2644/TestrepoIs.git>
2. Git Bash commands:

* git help : it provides frequently used several commands
* git help init : open documentation of particular command
* git init : It is used to create empty repository or reinitialized existing repo
* git status : This command will display status of current repository , here you can see

staged file:

* files which are added by git and ready to commit
* These file display in Green color
* Example : Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: Demo.java

un staged file:

* modified files will be display here and we need to stage these files to commit
* These file name will display in red color
* Example :

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: Demo.java

untracked file:

* newly created files will be considered as untracked file, we need to stage them to commit with help git add command
* These file names display in red color
* Example : Untracked files:

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

Demo.java

1. git add:

* git add filename: this command is used to add file in stagging area
* git add - - a : It is used to add all untracked file at a time

$ git add --a ClientApp.java Test.java

1. git rm: This command to untracked or un-staged newly created files

git rm –cached <file name >

git rm –cached \*: it will untracked all file

1. git commit: This command is used to commit files from working directory to local repository

git commit -m “commit massage”

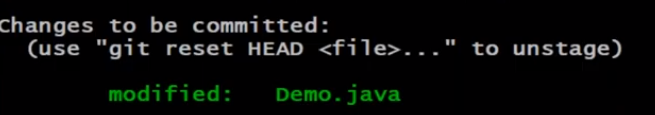
Note:

* git commit will be available for local repository (machine)
* when you execute git commit, all files will be in staging area and ready to push in central repo
* if you want to commit few files not all the files which is in staging area, that is not possible, you must untracked files which you don’t want to push in central repository

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*To commit our files to remote repo we should use below commands \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. git remote add <repo-URL> : this command require only time to connect your local repository to central repository
2. git push -u origin master : This is used to move changes from local repository to central
3. git reset : it is used to un stage a file

git reset head <file name>



1. git checkout : it used to remove all the changes done in the file

git checkout -- <file name >

git checkout -- \*

Yesterday’s session: GIT HUB Introduction:

* Git hub is cloud platform where we can store our project source code
* Git HUB uses git as version control software
* In GIT Hub we will create repository for storing our source coe
* All developers will install git client software to interact with GIT HUB repository
* Repository: It is a place where we can keep our code
* In git hub we can create two repository, they are public and provide repository
* Public repository: everybody can repository and we will decide who will commit
* Private repository: we can decide who can see repo and who can commit it
* Git client: it is software used to interact with git hub repository
* Git bash: it is cli tool to perform GIT operation
* GIT commands that we used so fart,

1. Init : to represent our folder as git folder
2. Status: It is used to show stage, un stage and untracked

* files which are added by git and ready to commit
* These file display in Green color
* Example : Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: Demo.java

un staged file:

* modified files will be display here and we need to stage these files to commit
* These file name will display in red color
* Example :

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)

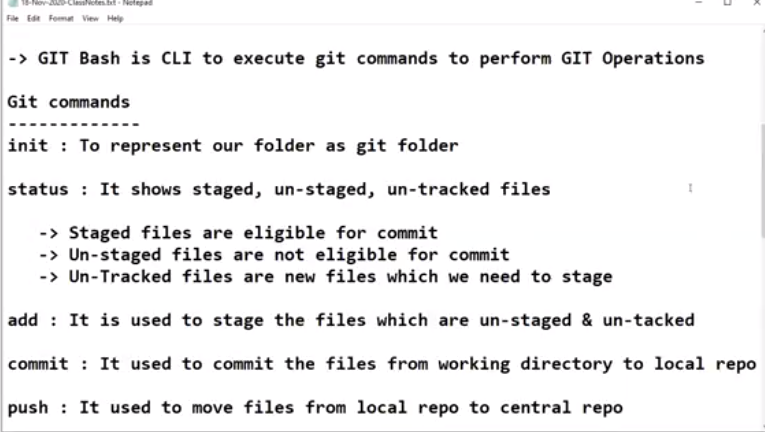
modified: Demo.java

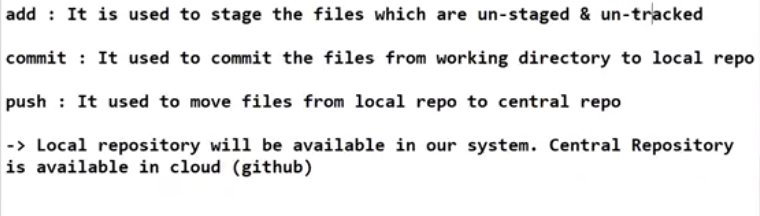
untracked file:

* newly created files will be considered as untracked file, we need to stage them to commit with help git add command
* These file names display in red color
* Example : Untracked files:

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

Demo.java





Today’s Session: Git Commands Practice

* When we make changes to files we need to stage them to commit
* To files to stagging area we will use add command
* To commit files to local repo we will use commit command along with message about commit
* To publish local commits to central repo we will use push command
* Whenever we commit, git will generate commit-id
* Commit ID contains 40 alphanumeric characters
* From 40 alphanumeric characters it will display first 7 character

# To check commit history we will use git log command

* sysntax: git log
* In commit logs it will display below details

1. Commit-id
2. Author
3. Timestamp
4. Commit msg

1. git clone : to take existing project from repository to local system we will use git clone command

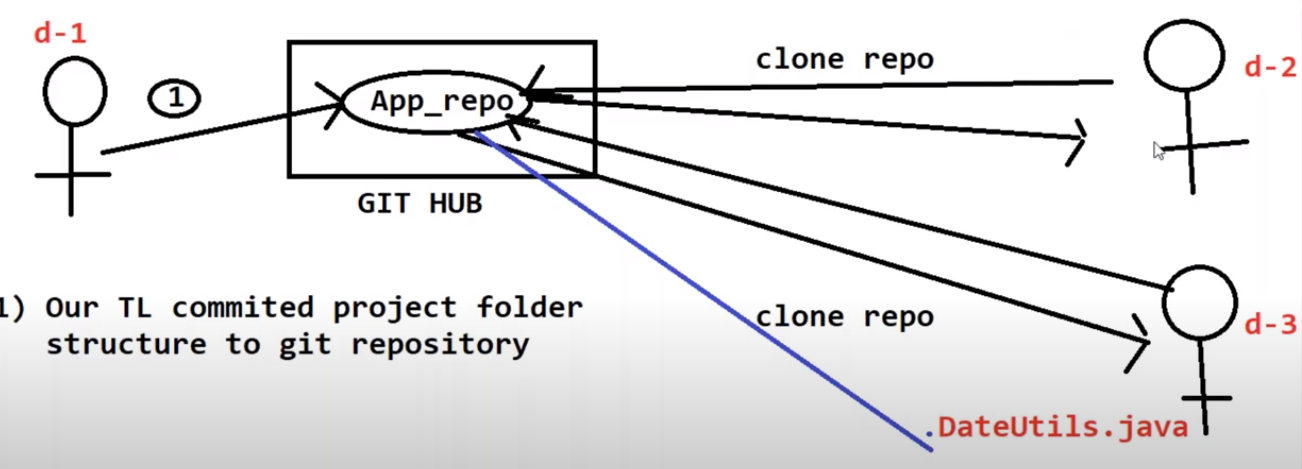
* syntax : git clone <Repo URL >

1. git pull :

* Note: Before making any changes to file in local repo, it is highly recommended to take latest changes from the central repo. To take the latest changes from the repository we will use git pull command
* Sysntax : git pull

When to use clone or pull?

Clone command is used to take the project source code first time and pull command is used to take latest upate from the central repository



Today’s session : Upload Spring project to repo

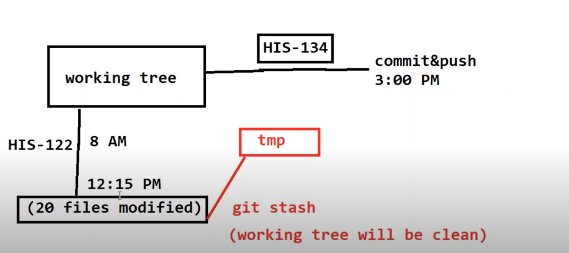
* Create repo in GIT Hub account
* Create spring boot application in STS IDE
* Got to workplace folder where project is created
* Inside the project folder open git bash
* Execute git init command
* Execute git add –a command
* Execute git commit -m “ message “ command
* Execute git remote add <Repo URL>
* Execute git push -u origin main command

**Note: Repository name and project name can be different, there is no relation**

**Note: file name which is specified in .gitignore that cannot be committed and push to central repo**

1. Git stash:

* It is used to record current changes and make working tree clean



Let’s consider, your TL assigned a new job today at 8 AM and you started working on it,

After 12:15 PM he assigned a new task which is more important then pervious task and he is telling you to do that job first, In this scenario, you have already worked on morning task and changed multiple files in the source code and it is difficult to revert all the changes and work on the new task,

In that case you can use git stash command to move all the changes files to temp and keep the source code as it when you changed it in the morning,

shinde7@DA18091945 MINGW64 ~/Downloads/Pradeep/JavaRealTimeRepo (main)

$ git stash

Saved working directory and index state WIP on main: 5a8a3d1 Added a new file b

y Ramesh Test2

shinde7@DA18091945 MINGW64 ~/Downloads/Pradeep/JavaRealTimeRepo (main)

$

**Git stash apply** : whatever changes you made in the morning that will be back and you can continue your work .

**Git stash clear** : This command will remove all the changes which you have done after the git stash

Today’s Session: Working with Branches in GIT HUB

What is branch in git HUB ?

Why we need branches in GIT repository?

How to create Branches in GIT repository?

How to merge branches in GIT repository?

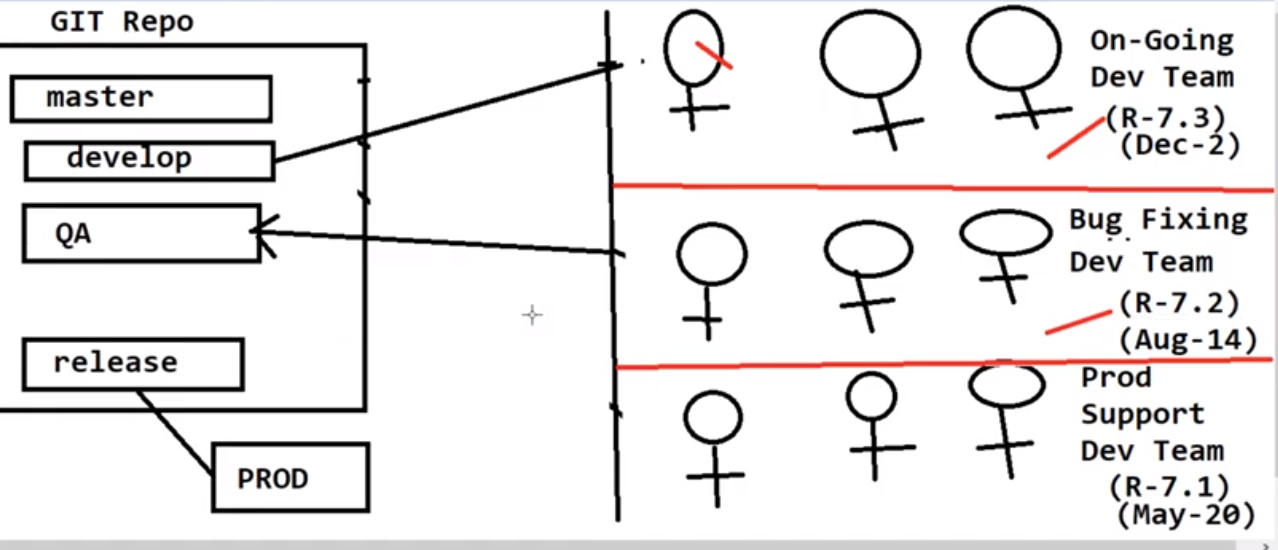
What is pull request?

How to resolve conflicts in the code?

What is branch in git HUB :

when we create GIT repository by default it will provide Master/main branch

* Branches are nothing but code bases
* We can create several branches in GIT repository
* Generally, in GIT repo we will create branches below like
* Master (default)
* Develop
* Feature
* QA
* UAT
* Release



Workflow:

* Login to GIT repo
* Create develop branch from the master
* Clean develop branch code

If you want to clone a specific branch code the we should execute below command:

Git clone -b <branch-name> <repo-url>

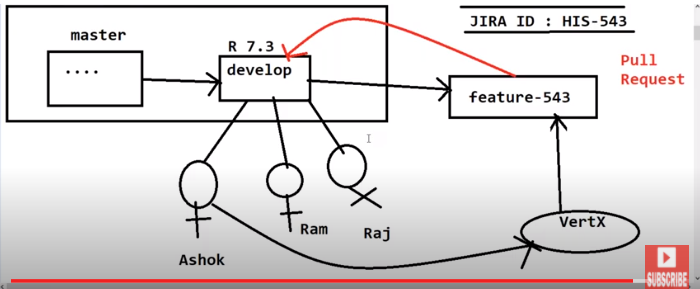
What is Pull request?

Merging one branch code to another with help of pull request, is called pull request

Let’s consider you’re working on a new functionality and to work on new functionality we need to crate a separate branch so new functionality code won’t affect to other branches,

Once you done with development of new functionality you can merge code from one branch to another this can be done with help of pull request and it is also called as branch merging

It may possible there is code confilicut while merging one branch to another.



28.12

# Junit:

What is software : The process of testing the code is called Software testing.

It helps to Identify the following

1. Incorrect implementation of the business rules
2. Weak software performance
3. Incorrect results of data searches
4. Incorrect file handling

How many ways we can do software testing?

Manual Testing : A test Engineer create test cases manually and executed it as per the client requirement, this process is called Manual testing, Test Engineer will paly a role of client to find bugs and wrong behavior of the software.

Automation Testing : Testing formed with the help of automation tool like, QTP , load Runner and JMETER etc

# Testing Stages:

* Unit Testing
* Integration Testing
* Smoke
* Functional
* Regression
* Performance

We will focus on Unit Testing,

Unit testing 🡪 Developer check the particular method in the project is working fine or not Instated of integration of complete code and deployment, This type of testing is called Unit testing

Integration Testing🡪 Once we completed with Unit Testing, Developer integrate and deploy code to test end to end , this type of testing is called is Integration testing

# Smoke Testing 🡪 (All screens are accessible or not)

Once these testing completed, Developer send project build to the tester team and they will deploy the build in the environment and first check all the screen redirection happening the project or not this process of testing is called Smoke testing, if there is screen redirection problem, They will send bug report to the developer.

# Functional Testing🡪

After completing Smoke testing, Tester run test cases to check the functionality such testing is called as Functional testing

# Regression Testing🡪

If Tester found any bug during functional testing, they will contact to the developer and developer fix the bug and deliver build to the Tester and Tester will do end to end testing again and also they will check what fix is implemented, is there any impact to other functionality due to new code such process of testing is called Regression Testing

# Performance Testing 🡪

Check the response time of the project, how many users can login at time such type of testing is called performance testing

# Why to do Unit Testing:

Unit testing is used to identify defects early in the software deployment

Proper unit testing done during the development stages to save time and money.

# Junit with Mocking🡪

What is unit testing : The process of testing

What is Junit?: It is open source framework, it is used to perform unit testing in java

Annotations available in the Junit?

@Test : it used to represent unit test method

@Asset:

@beforeClass

@After