MASTER IN

22 December 2022

19:43

|  |  |  |  |
| --- | --- | --- | --- |
| **SNO** | **TOPIC** | **NOTES** |  |
| 1 | GIT | <https://gitlab.com/>  Username - pradeep.viswa@gmail.com  Password - Infy@123  <https://gitlab.com/mygroup2280/MyProject.git> | cd existing\_repogit remote add origin <https://gitlab.com/mygroup2280/MyProject.gitgit> branch -M maingit push -uf origin main |
|  | Edureka | <https://learning.edureka.co/mycourses>  Username - pradeep.viswa@gmail.com  Password - Infy@!23 |  |
|  | AWS | <http://aws.amazon.com/>  Username: pradeep.viswa@gmail.com  Password: Infy@2646187 |  |
|  | GitHub | <https://github.com/>  Username: pradeep.viswa@gmail.com  Password: Infy@2646187  Token:  ghp\_QHng0nLUcjVMBqWVgBhNWg3Rqklo1G19f8tK | <https://github.com/pradeepviswa/myproject.git> |

Edureka Class - DevOps Training

16 January 2023

10:34

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  |  | | Hello Pradeep Viswanathan, |  | |  |  | | Thank you for registering for DevOps\_16-Jan-2023\_20\_30. You can find information about this meeting below. |  | |  |  | | |  |  |  | | --- | --- | --- | | **DevOps\_16-Jan-2023\_20\_30** |  |  | | Date & Time | Jan 16, 2023 08:30 PM Mumbai, Kolkata, New Delhi |  | |  | Jan 17, 2023 08:30 PM Mumbai, Kolkata, New Delhi |  | |  | Jan 18, 2023 08:30 PM Mumbai, Kolkata, New Delhi |  | |  | Jan 19, 2023 08:30 PM Mumbai, Kolkata, New Delhi |  | |  | Jan 20, 2023 08:30 PM Mumbai, Kolkata, New Delhi |  | |  | Jan 23, 2023 08:30 PM Mumbai, Kolkata, New Delhi |  | |  | Jan 24, 2023 08:30 PM Mumbai, Kolkata, New Delhi |  | |  | Jan 25, 2023 08:30 PM Mumbai, Kolkata, New Delhi |  | |  | Jan 26, 2023 08:30 PM Mumbai, Kolkata, New Delhi |  | |  | Jan 27, 2023 08:30 PM Mumbai, Kolkata, New Delhi |  | |  | Jan 30, 2023 08:30 PM Mumbai, Kolkata, New Delhi |  | |  | Jan 31, 2023 08:30 PM Mumbai, Kolkata, New Delhi |  | |  | Feb 1, 2023 08:30 PM Mumbai, Kolkata, New Delhi |  | |  | Feb 2, 2023 08:30 PM Mumbai, Kolkata, New Delhi |  | |  | Feb 3, 2023 08:30 PM Mumbai, Kolkata, New Delhi |  | |  | Feb 6, 2023 08:30 PM Mumbai, Kolkata, New Delhi |  | |  | Feb 7, 2023 08:30 PM Mumbai, Kolkata, New Delhi |  | |  | Feb 8, 2023 08:30 PM Mumbai, Kolkata, New Delhi |  | | Meeting ID | 944 7108 7660 |  | | Passcode | wK&fSxe6 |  | |  |  |  | |  | |  |  | | [Add to Calendar(.ics)](https://edureka-co.zoom.us/meeting/attendee/tJApcOiopjwsHNSXu1lpPcoyJiHLuR7XmaNF/ics?user_id=FvdFVATzp1_cRb1sQm7mAqG-K5x26G2TEr6mjgtzaCvOF03LApY.NEGyQjaeOxNfXgv4&type=icalendar)  |   [Add to Google Calendar](https://edureka-co.zoom.us/meeting/attendee/tJApcOiopjwsHNSXu1lpPcoyJiHLuR7XmaNF/calendar/google/add?user_id=ZYiTRcj-LUfLoYE0gcFHWAmxovtMdrzmNJpk_K5dRMb27OrvMXo.AduwwIjBZwdW6Bb0&type=google)  |   [Add to Yahoo Calendar](https://edureka-co.zoom.us/meeting/attendee/tJApcOiopjwsHNSXu1lpPcoyJiHLuR7XmaNF/cal?user_id=DF8UF2MmobjdJhd_jWXsXF_oqdbVy1DCKmBWa3KYymZnCNHZ_Gg.CdIWsqHMORQi5YKM&type=yahoo) |  | |  |  | | You can [cancel](https://edureka-co.zoom.us/meeting/register/tJApcOiopjwsHNSXu1lpPcoyJiHLuR7XmaNF/success?act=cancel&user_id=sL9UzoSeSR-lCgCJ1CnhuA)your registration at any time. |  | | Please submit any questions to: 515926@ISV.edureka.com. |  | | **WAYS TO JOIN ZOOM** |  | | **Join from PC, Mac, iPad, or Android** |  | | [Join Meeting](https://learning.edureka.co/mycourses) |  | | If the button above does not work, paste this into your browser: |  | | <https://learning.edureka.co/mycourses> |  | | **To keep this meeting secure, do not share this link publicly.** |  |     Thank you! |

Day 01 - 16 Jan 2023 (Introduction)

16 January 2023

20:32

|  |  |
| --- | --- |
| AWS account | Email: pradeep.viswa@gmail.com  Password: Infy@2646187 |

Introduction to DevOps

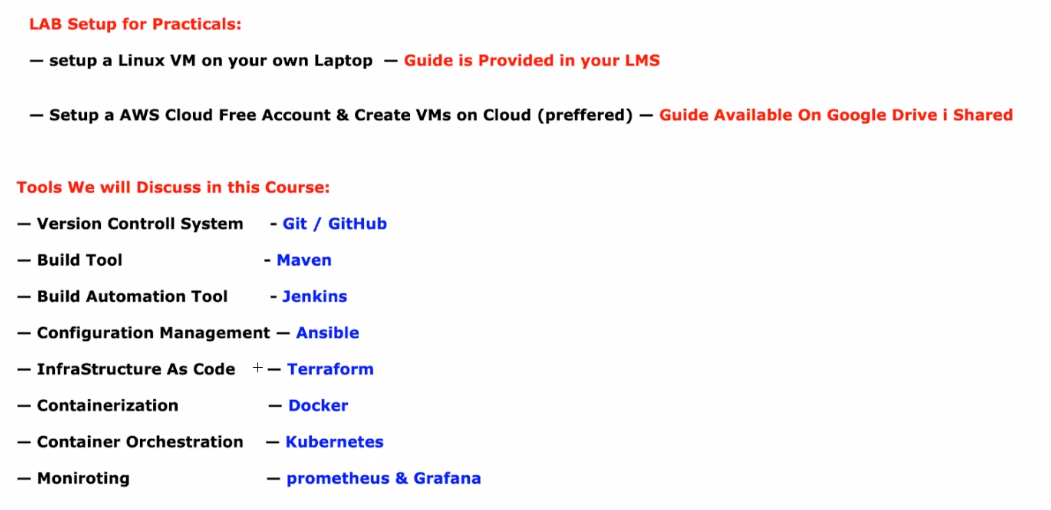
<https://learning.edureka.co/classroom/recording/1414/11756/1572552?tab=ClassRecording>

Day 02 - 17 Jan 2023 (Git)

17 January 2023

20:39

<https://learning.edureka.co/classroom/recording/1414/11757/1573007?tab=ClassRecording>



Today's Agenda:

1. Setup EC2
2. GIT

<https://aws.amazon.com>

Create new account

Login

1 Create virtual server

1. Services -> compute -> EC2 (virtual server in the cloud)

(To login that virtual machine remotely need some securitysetting)

1. Network & security-> security groups -> create a new security group   
   group name: edujan16  
   no change in VPC  
   **inbound rule ->**   
   add rule -> SSH, TCP, 22, source anywhere Ipv4

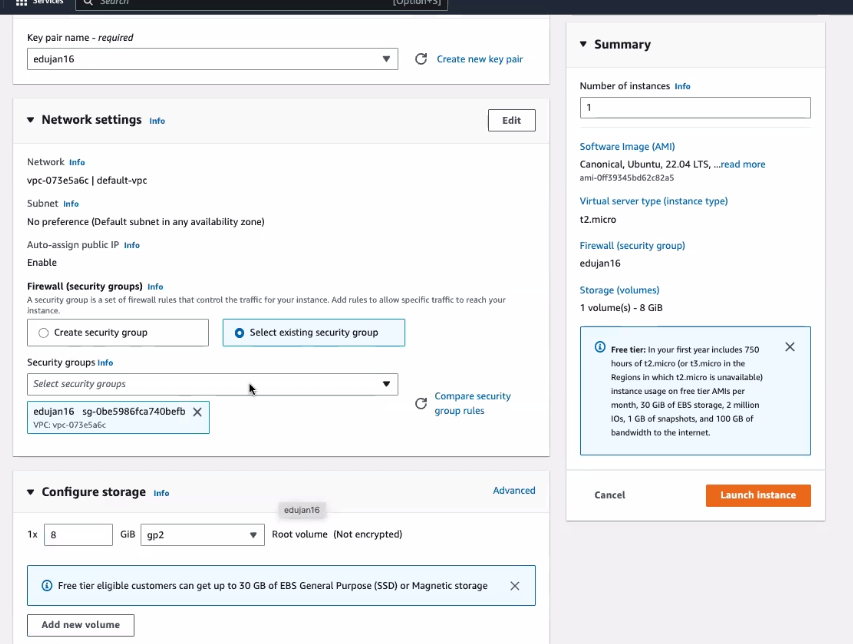
Add rule -> all trafic, All, All, Anywhere IPv4

**Outbond rule** -> let it be default

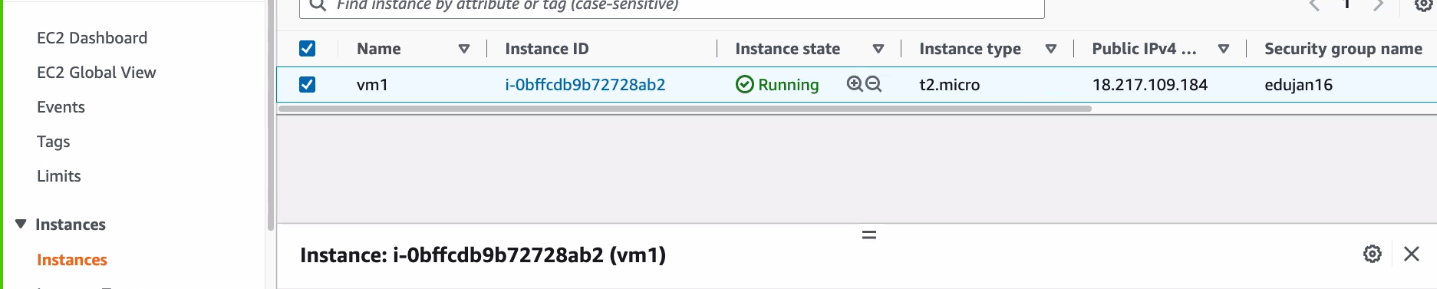
**Tags** is optional  
name -> edujan16  
  
create security group

Set credential for server  
Network & Security -> key Pairs  
Create key pair button  
edujan16  
RSA  
.pem  
CREATE A KEY PAIR  
  
this downloads file in system  
  
edujan16.pen file is downloaded in local computer

Instances (used to create a virtual machine)  
Launch Instance  
Name: vm1  
OS Image: ubuntu  
Select ubuntu 22.0.4 LTS (**free tier eligible**)  
64-bit (x86)  
Instance type: t2.micro (free tier eligible)  
Key Pair (Login): select from drop down and select **edujan16** which we cretaed  
Network settings: (firewall) select existign security group, from drop dwon select **edujan16** select  
Configure Storage: let it be default  
  
**review and click on launch instance**



view all instances  
or go to Instances -> Instances



**Topic: How to we login to Virtual Machine?**

ssh protocol

Need ssh client ool

Windows User:

Gitbach

Mobaxterm

Putty

Powershell / cmd

Mac user:

Mac terminal

To connect we need

Need public IP

Default Username: ubuntu

Password: \*.pem

On mac

ssh -i \*pem username@public-ip

chmod 400 edujan.pem

(it is basic permission)

ssh -i edujan16.pem ubuntu@18.217.6.5

Yes

Use mobaXXterm, it is similar to MAC

**VERSION CONTROL**

Called as

VCS - version control system

SCM - source code management

Types of VCS:

Local CS (SCCS) - very old

Centralized VCS (svn cvs, tfs)

Distributed VCS (Git/Github/bitbucket)

**Local VCS** -

DEV laptop/desktop

|  |
| --- |
| MYPROJECT /  FILE1 V1, V2  FILE2 V1, V2    SCCS - VSC TOOL |
|  |

**Centralized VCS** -

**Remote Server**

|  |  |
| --- | --- |
| Myproject/  File1 v1, v2  File2 v1, v2     SVN - Version Control tool |  |

**Distributed VCS**

|  |
| --- |
|  |

**Practical**

Install Git in your linux machine - ubuntu

How to install?

Command line installation: <https://github.com/lerndevops/labs/tree/master/git/install>

Practice Linux Commands: <https://drive.google.com/file/d/12YvQgxNew8q-WMPxp9BL5Pn--SJTujdA/view?usp=share_link>

**Install Using Script**

sudo wget <https://raw.githubusercontent.com/lerndevops/labs/master/scripts/installGit.sh> -P /tmp  
sudo chmod 755 /tmp/installGit.sh  
sudo bash /tmp/installGit.sh

**Manual Installation Steps**

**Install Git on - Ubuntu OS**

sudo su -  
sudo apt-get install software-properties-common  
sudo add-apt-repository ppa:git-core/ppa -y  
sudo apt-get update  
sudo apt-get install git -y  
git --version ( if it writes the version then the installation is successful )

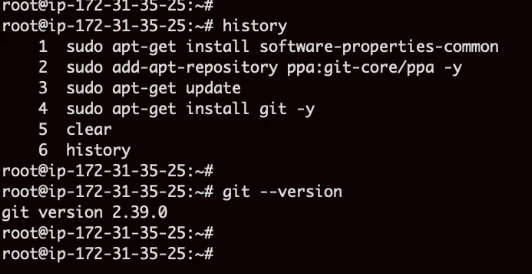
**Install Git On - CentOS / Fedora / RedHat / amzn Linux**

sudo yum install yum-utils  
sudo yum update  
sudo yum install git

From <[*https://github.com/lerndevops/labs/tree/master/git/install*](https://github.com/lerndevops/labs/tree/master/git/install)>

**Apt-get** s advanced Linux command to manage package

It allows to search for install, manage, update or remove software



Local repository where files will be stored to be uploaded in Git Hub

mkdir myproject

Cd myproject

Git init ##to initiralize the local repo

Initialzed empty git repo

ls -al

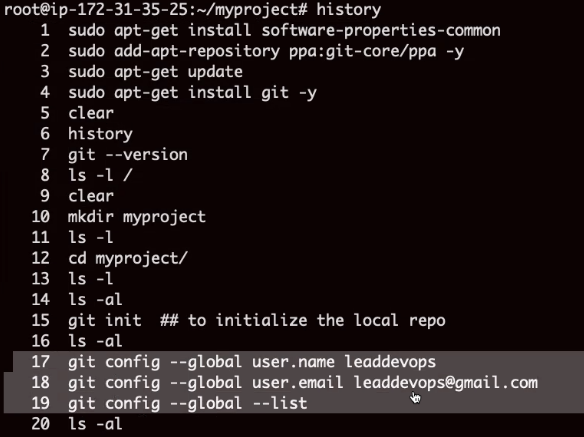
.git folder

##tell git who you are. Keep same name what you created in github.com portal

git config --global user.name **leaddevops**

git config --global user.emal **leaddevops@gmail.com**

git config --global --list



a

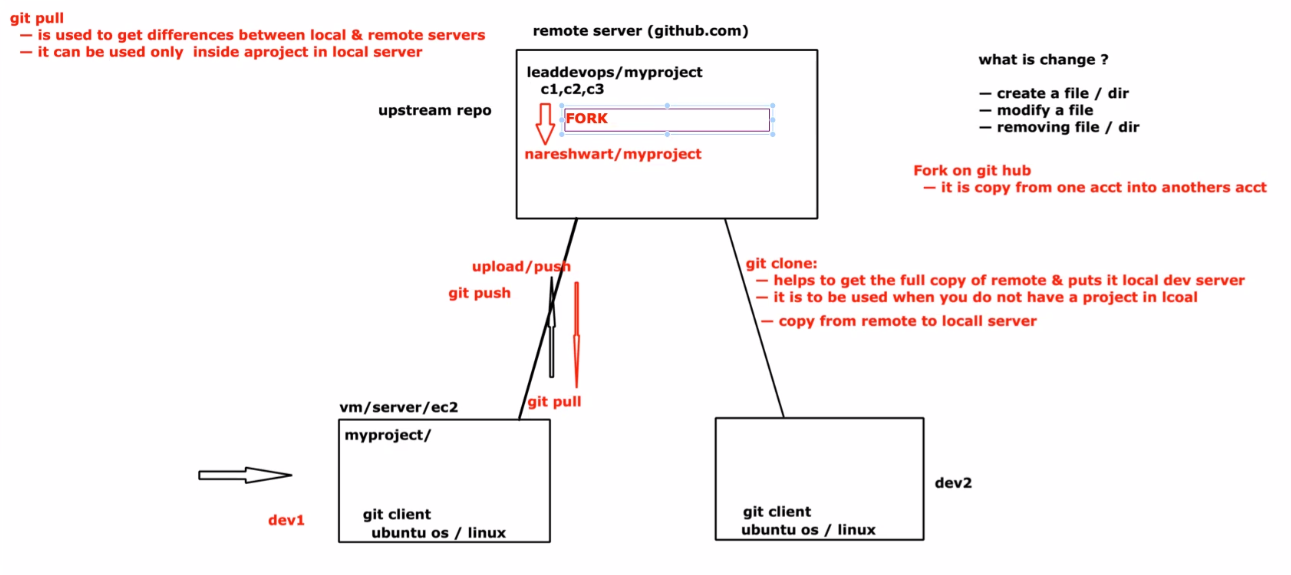
Day 03 - 18 Jan 2023 (Git)

18 January 2023

20:30

<https://learning.edureka.co/classroom/recording/1414/11758/1573764?tab=ClassRecording>

**Today's Agenda:**

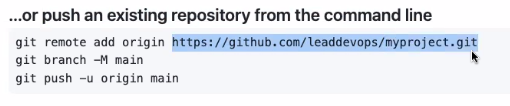


* 1. GIT

**New commit in Local**

* 1. git status
  2. Empty repo as of now. Create a new file
  3. touch a.java
  4. touch b.java, c.java
  5. git status *## files will appear under* ***untracked*** *files*
  6. git add a.java b.java ## track and b java
  7. git status *## files willa ppear under change to be committed*
  8. git add **.** *## track all changes I made to this dir*
  9. git commit -m "first commit" *## commit to local repo*
  10. git status ## no file will appear
  11. git log *## will show commit ID*
  12. vi a.java  
      i to edit  
      Esc, :wq
  13. git add a.java
  14. git status
  15. git commit -m "second commit"
  16. git log *##2 commit IDs will appear*
  17. git show <Commit ID> *## shows what was committed*
  18. rm now.java
  19. git add .
  20. git commit -m "deleted now.java"
  21. git log *## check 3rd commit*
  22. *We can revert only recorded change*
  23. git revert <commit ID> *## Ctrl+S to come out of screen*
  24. git log shows all commit IDs. Way to filter the log  
      git log --oneline
  25. git log --oneline -n 3 *## last 3 commits*
  26. git log --oneline --since <date> --after <date> ##two dates
  27. *git --help for full help*
  28. ***Version Control*** *git log --oneline  
      Many commits we can give one version number*
  29. *git tag -a "v1.0" <commit ID or last 7 digits> -m "release v1.0"*
  30. **
  31. Create multiple files and multiple commits
  32. git tag -a "v1.5" 369874ad -m "release v1.5" ##Tag again
  33. Tags are temporary metadata, temporary reference to commit
  34. Git tag -d v1.5 #delete a tag
  35. Git reset --hard <commit ID> *##Delete commit id*. VERY DANGEROUS
  36. Reset will delete commit ID which cannot be called again
  37. Revert creates a new commit ID

**New commit on Remote**

* 1. Account should be there on <https://github.com>
  2. Create a repository
  3. + icon, New Repository
  4. Name = **myproject**
  5. 
  6. Git remote -v
  7. Git remote add origin <https://github…./….git>
  8. Git remote -v
  9. Taken care by dev team
  10. git branch *## by default branch is master*
  11. Git branch -M main *## will rename branch master to main*
  12. Git push -u origin main *##used to upload local repo to remote repo. Upstream is remote.   
      origin is alias to* [*Https://…*](Https://…) *URL*
  13. Use authentication token instead of password
  14. How to get github token  
      <https://github.com/lerndevops/labs/blob/master/git/how-to-create-github-access-token.pdf>
  15. Github.com -> Account -> settings (in new tab)

|  |  |
| --- | --- |
| * + 1. developer settings (Bottom)     2. Personal access token     3. Generate a new token     4. Note = **edujan16**     5. Expiration = no expiration (not recommended)     6. Select all checkboxes     7. Generate a new token     8. Copy token, it won't be able to copy again |  |

* 1. Git push -u origin main  
     login ID: pradeep.viswa  
     password: **ghp\_AeNG0melDH5Xxiuxx3l5Cyeh0hkdiH26qBEV**
  2. Refresh github page
  3. Branch name and files should appear
  4. It shows commits as well
  5. Touch 1.java
  6. Git add 1.java
  7. Git commit -m "1.java"
  8. Git push -u origin main  
     username  
     password = <token>
  9. We can avoid passing credential all the time  
     store the credential to project itselt
  10. Git config --global credential.helper store
  11. Touch 2.java
  12. Git add 2.java
  13. Git commit -m "2.java"
  14. Git push -u origin main  
      username  
      password: <token> only one time. Will be saved
  15. Touch 3.java
  16. Git add 3.java
  17. Git commit -m "3.java"
  18. Git push -u origin main  
      <no cred will be asked for>

**New Dev joined the project, got new Dev computer with git client installed**

* 1. <https://Github.com> -> Code -> copy git path
  2. git clone https://.....
  3. cd myproject *## you can see all the code downloaded from remote*
  4. If repo is public, it doesn't require any authorization

**Dev2 wants to commit something  
PUSH**

* 1. Touch dev2.java
  2. Git add .
  3. Git commit -m "dev2"
  4. Git config --global credential.helper store
  5. Git push -u origin main  
     username  
     password
  6. Refresh <Https://github.com> repo page

**Dev1 wants to synch, dev2.java is missing  
PULL**

* 1. Git --oneline
  2. Git pull *## Only want to pull changes which are missing. This command can be triggered from inside project folder. i.e., myproject*

**FORK**

* 1. Logged in as leaddevops user  
     what if woner removes that project  
     before owner removes it, I wantto add it in my accuunt  
     open <https://github.com/leaddevops/myproject>
  2. 
  3. Creates a new project within my account
  4. The author can see who has FORKed the project
  5. Lab repo  
     <https://github.com/lerndevops/lab>
  6. Tag which was created in local
  7. From Dev1  
     git pull  
     git log --oneline  
     git push -u origin main --tags  
       
     **How to remove saved token?**
  8. git config --global --unset credential.helper

Practice

19 January 2023

17:09

Connect EC2 Instance

ssh -i edujan16.pem ubuntu@18.181.249.89

Github token: ghp\_QHng0nLUcjVMBqWVgBhNWg3Rqklo1G19f8tK

Day 04 - 19 Jan 2023 (Git)

19 January 2023

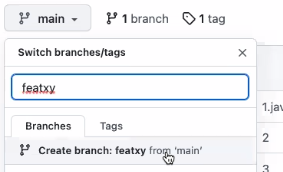
20:36

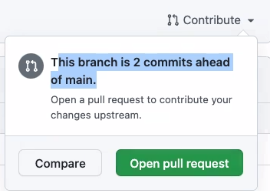
<https://learning.edureka.co/classroom/recording/1414/11759/1574314?tab=ClassRecording>

**Agenda**

* 1. GIT - Branching
  2. GIT - Merge

**GIT - Branching**

* 1. Lock MAIN branch for security reason.
  2. Git push -u origin main  
     all up to date
  3. Lock
     1. Settings
     2. Branches
     3. Add branch protection rule button
     4. Branch name: main
        1. Check require a pull request before merging
        2. Do not allow bypassing the above settings
  4. Trouch today.java
  5. Git add .
  6. Git commit -m "today"
  7. Git push - u origin main  
     **error:** changs mush be made through a pull request.  
     failed to push some refs to https://...
  8. Create a branch
     1. Main
     2. 

* 1. On dev laptop, local  
     **FETCH**
     1. Git branch  
        \*main
     2. Git pull  
        this will show new branches but they are not initialized on local
     3. Git **fetch** origin **featxy**:featxy  
        #this will replicate remote **featxy** branch with a newly created local branch called **featxy**.
     4. # Switch to featxy  
        git switch featxy
     5. #check active branch  
        git branch
     6. #Both branches should show same output  
        git log --oneline
     7. Touch featy.java
     8. Git add .
     9. Git commit -m "featy"
     10. Git log --oneline  
         see the difference. Check featxy branch
     11. #push the change to remote repo  
         git push -u origin featxy  
           
         **GIT - Merge (GUI Based only)**
  2. Changes made in featxy branch merge it in main.  
     can't be done directly, it requires PULL request
  3. Open Pull Request from **Contribute** option
  4. 
  5. Owner will be notified, check number of commits tab, number of files, add review comment
  6. Click on **Merge pull request** button  
     **Close pull request** means rejectign the request  
     *(need more practice here)*

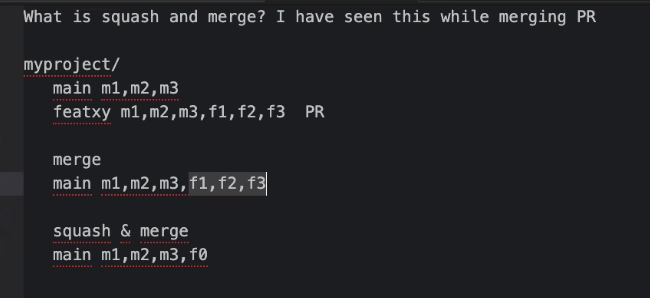
*TEST BY ADDING IN GITHUB DIRECTLY AND THEN PULL REQUEST ON LOCAL*

* 1. We can create local branch  
     git branch **hello**git checkout -b hello *#branch will be created and switched*
  2. Delete branch on local  
     git branch -d hello  
     git branch -D hello *#forece delete*
  3. *Merge on local  
     git switch featxy* ***#first switch to*** *Git merge main*

In github.com

Settings -> collaborator -> add people





**MERGE CONFLICT and how to fix it**

* 1. Multiple developer working on same feature
  2. Dev1 & Dev2 machine
     1. Git fetch origin featxy:featxy
     2. Git pull  
        both up to date
  3. Dev1 machine
     1. Vi featx.java  
        hi there from dev1
     2. Git add .
     3. Git c ommit -m "mod featx dev1"
  4. Dev2 Machine
     1. Vi featx.java  
        hi there from dev2
     2. Git add .
     3. Git commit -m "featx dev2"
  5. If both added same code, duplicate
  6. Dev1
     1. Git push -u origin featxy
  7. Dev2
     1. Git push -u origin featxy  
        will fail
     2. Git pull   
        auto-merging featx.java  
        CONFLICT content. Merge conflict in faeatx.java
     3. Vi featx.java  
        dev1 and dev2 code of lines will appear here  
        review it and save it
     4. Git status #both modified
     5. Git add .
     6. Git commit -m "checked conflict"
     7. Git push -u origin featxyrevert