Saturday, August 2, 2025

GIT - SOURCE CODE MANAGEMENT

1. REVIEW SAMPLE ONLINE PROFILE AND HOW TO BUILD YOURS MY JOB PROFILE

REVISION OF WHAT WE COVERED LAST WEEK

Any challenges from last class we should investigate?

Make sure the following is installed

1. Git bash
2. Vscode
3. GitHub account created

REVISION OF WHAT WE COVERED LAST WEEK

Revision of last week.

1. Git is a distributed versions control system. [dvcs]
2. It is used to track changes to files [there it records all changes]
3. It is widely used by software developers and IT teams .
4. It professionals use git for collaboration.

HOW TO START A GIT PROJECT

PLATFORMS

1. VSCODE INSTALLED AND GIT PLUS YOUR GITHUB ACCOUNT

STEPS	Direction	Examples
1.	Create a project directory	open vs code and follow instructions
2.	Create files to be tracked	See video
3.	Start git	# git init
4.	Add contents in the file	
5.	# check tracking status	# git status
6.	# add files to the staging area to be committed	# git add.
7.	Git identification	git config - -global user.name “prof” # git config - -global user.email “prof@liontech.com”
8.	Git commit /save your changes	# git commit -m “commit message”
9.	Add your remote repo link to the local	# git remote add origin https://github.com/CLASS32-DEVOPS-cloud/class32-collaboration-repo.git
10.	Push to the repo	# git push origin master

Git authentication.

URL TYPE	APPROACH/ authentication method
# https	PAT: personal access token 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
# ssh	# keypairs /private and public 1. Run this command to create two keypais # ssh-keygen 2. Access the .ssh directory to access keys # cd ~/.ssh 3. Display the content of the key # cat id------.pub 4. Copy the content of the key

Git identification

 # git config - -global user.name “prof”

 # git config - -global user.email “prof@liontech.com”

15 minutes break

GIT COLLABORATION AND GIT BRANCHING

A. Sent invitations to team members with write access attached so that team members can collaborate with having permissions.

Collaboration on the repository using the concept of PULL REQUEST

1. Introduction to Pull Requests

A pull request [PR] is a mechanism for proposing changes in a Git repository – allowing team members to;

1. Review
2. Discuss
3. Approve modifications.

KEY BENEFITS OF PRS

1. Code review: Ensure quality and correctness
2. Collaboration: Encourage discussions and feedback
3. Documentation: provide a history of changes and reasoning.
4. CI/CD Integration: Automated testing before merging

HOW TO COLLABORATE ON AN EXISTING REPOSITORY USING BRANCHING AND PULL REQUEST

GIT COMMANDS	EXPLANATIONS
# git clone git@github.com:Hands-onclass32CFL/CLASS32-PUBLIC-REPO.git	**Clone the repository u want to collaborate**
2. Perform the required changes	**Perform changes on files**
3. # git add . to confirm changes	
4. Create a git branch	**# git branch prof** **# git branch eric** **# git branch nicholas**
5. Switch to the git branch	**# git checkout prof**
6. Commit your changes on your new branch	**# git commit -m "message"**
7. Push your changes to the your branch	**# git push origin prof**
8. Create the pull request on the browser and assign reviewers	

Sunday, August 3, 2025

DEVOPS AND CLOUD COMPUTING TRAINING

SOURCE CODE MANAGEMENT

SUMMARY OF STARTING A LOCAL GIT PROJECT

1. Create a project directory /folder [mkdir mtn-momo-app]
2. Access the project you just created [cd mtn-momo-app]
3. Create your project files [touch payment.txt]
4. Start the local repository [git init]
5. Check your file status – whether is being tracked or not [git status]

BEFORE GIT COMMIT – IDENTITY YOURSELF

```
#  git config  - -global user.name  “akogu”
# git config  - -global  user.email  “demo@demo.com”
```

6. Git commit [git commit -m “your message”]
7. ADD YOUR REPOSITORY URL

```
# git remote add  origin  [url of repo]
```

GIT AUTHENTICATION

TWO METHODS OF AUTHENTICATION WITH GIT .

1. **SSH URLS [KEYPS - PUBLIC AND PRIVATE KEY]**

a. **Create two keypairs – public and private and upload the public to your github account . [ssh-keygen]**
b. **Upon creation your key is by default saved in .ssh directory**
c. **To access the key – run [cd ~/.ssh] then run [ls -lt]**

2. **HTTPS [PERSONAL ACCESS TOKEN – ON HOW TO CREATE SEE YOUR VIDEOS]**
8. **Push your local repo to remote**

 # git push origin master

LIVE DEMO ON LOCAL REPOSITORY - START AND PUSH TO GITHUB

COLLABORATING ON A GIT REPOSITORY

GIT PULL – GIT CLONE – FORK AND PULL REQUEST

Git clone: to clone an existing means to
 # git clone [url]

1. **Download the content of the remote repositor / directory with all the files to your local computer. c**

 # Only clone a particular once .

Some key facts about git clone

1. **Anyone can clone a public repository and read**
2. **Only those with write access can propose changes via pull request / merge request**
3. **You wont be able to clone a private repository unless you have been invited or you have write access to the repository.**

GIT PULL COMMAND

1. **Git pull fetch /pulls changes in the remote that is not available on your local and brings ur local up to date .**
2. **Git pull helps in avoiding merge conflict.**

HANDS ON - GIT CLONE – PULL AND PULL REQUEST.

THIS IS A JOINED PROJECT

1. Clone the repository

git clone git@github.com:CLASS32-DEVOPS-cloud/august-repo.git

2. Access the repository [cd or use vs code to open it]
3. Perform changes
4. RUN git add .
5. Create a branch

git branch prof

6. Switch to the branch

git checkout prof

7. Commit your changes to branch prof

git commit -m “your message”

8. Push to branch prof

git push origin prof

9. Create pull request and assign reviewers