02. Git commands.

* Install latest version of Git and Go to > programfiles > git > cmd
* copy link and paste at System Variables (path)
  C:\Program Files\Git\cmd.

commands:

$ git config --global user.name " "
$ git config --global user.email " "
$ git --version
$ git --help.

(i) create a folder; add files into the folder
Run cmd from the folder; follow steps

1) $ git init

Initialized empty git Repo in C:\user\ ...

2) $ git status

On Branch Master
No commits yet
untracked files
Kamal.txt
Kamal.java.

3) $ git add . (for all files adding) (or)
$ gid add. Kamal.txt

4) $ git commit

on branch Master
nothing to commit

5) $ git commit -m "first change"

6) $ git branch -M master

7) $ git Remote add origin git@github.com:KAMALESWAR06/Repo.git

8) $ git push -u origin master

Enter credentials and
files are pushed successfully.

Source code Managment:

9) Fork the Repo (others)

10) $ git clone git@github.com:KAMAL06/Repo-git (Cours)

11) $ git branch -M branch1

12) $ git add .

13) $ git commit -m "branching"

14) $ git push --set-upstream origin branch1

new branch are created
and file have been pushed

15) create pull request

16) Merge pull request.

04) Jenkins Installation and setup :

01) Install latest version of Java (JDK)
>> JDK 22 X64 Installer for windows

02) After installing Java Set environment Vasiable for jdk.

JAVA_HOME : C:\profiles\Java\jdk-
path : C:\profiles\Java\jdk-22\b

03) Dowload Jenkins LTS (.war) file.

04) In cmd enter

java -jar C:\Jenkins.war --httpPort=
Jenkins path.

If any error for Java version try, s

$ java -jar jenkins.war --enable-future-jav
--httpPort = 9090.

$ java -jar jenkins.war --httpPort = 9090

05) Navigate to chrome and enter
http://localhost:9090.

06) unlock Jenkins : Administration Password in cmd.

07) Install Selected plugins and login with credentials

08) Jenkins Installed and running Successfully

6i) Freestyle project with Git Hub :

i) open jenkins from chrome with localhost :

ii) New item > Jenkins > freestyle project.

iii) General :> Github project :
Project url : Github link

iii) Sourcecode Management : Git
Repo url : Github .git link.

iv) Build Triggers

v) Build Environment

vi) Build Steps : Execute Windows Command

```
$ javac Kamal.java
$ java Kamal
echo " Hello Kamal "
```

SAVE

vii) post build actions

Build Now ✓ #1, 15 may 2024

console output : i) Java Kamal > Hello world

ii) echo : Hello Kamal

```
stage ('Test') {
    steps {
        echo "Testing project"
    }
}
stage ('Release') {
    steps {
        echo "Releasing project"
    }
}
}
```

4) Build Now

Build ✓ #3

Pipeline Job have been created

* pipeline is a series of events that are connected together to make quick SoftwareReleas
* pipeline is responsible for building codes, running tests, deploying new versions
* CI is a practice that integrates code into Repo and finds the bugs which are done automated
* CD is the phase in which the changes are made to the code before deploying.

06) Docker commands

Install Docker Desktop on your windows.

Enable 1) Hyper-V
2) wsl (windows subsystem for linux)

in cmd :
i) wsl --install
ii) wsl.exe --install --no-distribution

Restart the computer and start Docker commands

1) docker --version

docker version 26.1.1 build 4cf5afa

2) $ docker run hello-world

Hello from docker

3) $ docker images

Repo	ID	SIZE
hellowrld	d2c36	13.3Kb

4) $ docker pull ubuntu -

4966384cc : pull complete
Downloads ubuntu image from Docker hub.

9) $ docker commit 13744a Kamaleswar/ubuntu
$ docker commit <con ID> <user name>/<imag>

Sha256 : 7df7542385015ba96.

i) creates a new image from container with container ID.

10) $ docker push Kamaleswar06/ubuntu.

The push c8da765b0 : pushed.

i) The command is used to upload a docker image from local to docker registry.

11) $ docker rm 13744a12f

Deleted container with id 13744a12f

12) $ docker ps -a

13) $ docker images.

14) $ docker rmi bf3dc08bfed0

Deleted : Image Kamaleswar06/ubuntu with id bf3dc08bfed.

10) Develop Test cases for above Selenium.

1) Same steps and commands as previous but the code changes internally.

Kamal1.js

```
const {Builder, By, Key, until} = require ('seleum-webdri')
async function runtests( ){
    let driver = await new builder ( ).forBrowser ('chr')
    try {
        await driver.get ('https://www.google.com');
        console.log ('Test case 1 : Google Success')
        let images link = await driver. findElement
        await imageslink .click( );
        :
    } finally { await driver. quit ( ); }
}
run tests ( ) ;
```

2) Run $ node Kamal1.js in cmd.

Test case 1 : Google page opened
Test case 2 : Search performed
Test case3 : clicked on Images
Test case4 : clicked on News
Test case5 : Back to google.

09) write a Simple js program and perform testing using selenium.

1) Download Nodejs latest version
2) setup path for Node JS from environment var
3) open cmd :

i) $ node js
ii) $ npm install selenium-webdriver
iii) $ mkdir Kamal
iv) $ cd Kamal
v) $ type nul > Kamal1.js

disc c → users → CSE → Kamal1 (copy code)

```
const {Builder, By, Key, until} = require('selenium-webdriver');

async function New1() {
    let driver = await new Builder().build();
    try {
        await driver.get('https://google.com');
        let title = await driver.getTitle();
    } finally {}
}
new1();
```

4) After coping code run the command, node Kamal1.js

$ node Kamal1.js (Directing to google

5) $ docker run -it -d ubuntu

13744a125ff3716576047b964d b24c1a

i) creats a new container from ubuntu image

ii) STDIN and tty are open (-i & -t) for command

iii) Run in the background (-d) detached

6) $ docker ps -a

CONTA ID	IMAGE	Status
13744a125	ubuntu	up 9 seconds

i) The command lists all currently running containers. (-a : all).

7) $ docker exec -it 13744a125 bash

i) Run command into present container

ii) 13744a125 - container ID

iii) -it (STDIN & tty)

iv) bash shell for executing

$root@13744a125 : / # echo Iam Kamal

Iam Kamal

$root@13744a125 : / # exit

exit.

8) $ docker stop 13744a125

13744a125

05) CI / CD using Jenkins :

1) In Jenkins create a pipeline project.

2) i) General → Nothing
   ii) Advanced options → Nothing
   iii) Pipeline : pipeline Script from scm
        SCM : Git
        Repo URL :
        credentials :
        Script path : jenkins01

3) Now Go to the Github repo which has been provided and create new file jenkins01 and insert code: (Script path is same as file name)

```
Pipeline {
    agent any

    Stages {
        Stage ('Build') {
            Steps {
                echo "Building project"
            }
        }

        Stage ('Deploy') {
            Steps {
                echo "Deploying project"
            }
        }
```