1. **Create AWS Instance and connect it to putty**

   Login AWS account -> search ec2 -> launch instance -> give name and create key pair -> and launch it -> select the instance created -> and copy the public Ip address

   Open putty -> paste the public Ip address -> In Connection(SSH -> Auth -> Credentials)m -> In private key file for authentication (Browse the key created for AWS Instance) -> accept

   Login ->ec2-user

2. **Install Docker on AWS instance and show the version of it**

   Follow same steps in Q1. then perform following commands

   Sudo yum update -y
   Sudo yum install docker
   Sudo docker --version

3. **Install git on AWS instance nd show the version of it.**

   Follow same steps in Q1. then perform following commands

   Sudo yum update -y
   Sudo yum install git
   Sudo git --version

4. **Compile project in Jenkins using Maven**

5. **Create slave node in Jenkins**

   In Jenkins(localhost:8080)

   Manage Jenkins -> In Security(go in security -> Agents(random) -> apply and save)

   Manage Jenkins -> In System Configuration -> Nodes -> New node(give name & select permanent agent) and create -> then

   Name: (name of the node)

   Description: (any description)

   Number of executors: 2

   Remote root directory: (for this create a new folder on desktop and paste that path)

   Labels: (give any name)

   Usage: keep default

   Launch Method -> Custom WorkDir path -> (give the same path which you have given in remote root directory)

Then save

After saving -> click on the node name which you have created -> in that look for (Run from agent command line: windows) -> you will get commands for windows -> open command prompt (run as administrator) -> and copy and paste the commands in your cmd ->then it will get connected.

6. **Assign project to slave node and build project**

   Follow same steps in Q5. then
   In Jenkins -> Dashboard -> create free style project (new item -> enter name and select ‘freestyle project’ -> Configure (in general -> select ‘Restrict where this project can be run) -> then in Label expression write the name of the node which you have created in Q5 -> then in build steps -> select ‘Execute Windows batch command’ -> there write (echo ‘hello world’) -> apply and save -> build now -> console output

7. **Run Selenium Tests in Jenkins Using Maven**

8. **Build pipeline project in Jenkins**

   In Jenkins -> Dashboard -> new item -> give name for pipeline and select ‘pipeline’ -> click OK -> Configure (go to pipeline -> in try sample Pipeline select ‘Hello World’) -> apply and save -> Build now and console output

9. **To perform various GIT operation on local and Remote repositories**

   Operations on local repository

   Create a folder on your desktop -> create a text file in it -> ex: file1.txt and in that write “hello world” and save

   Open that folder in command prompt and execute following commands

   git init

   git status

   git add file1.txt

   git commit -m "file1 added"

   git log

   git checkout -b new_branch

   git checkout master

   git merge new_branch

   git checkout -- file1.txt

   git rm file1.txt

   git commit -m "Removed file1.txt"

Operations on remote repository

(Make a github repo -> create file -> enter some text (ex: “hello world”))

git clone https://github.com/saishrane-11/demorepo.git
cd demorepo
(Before pulling -> go in repository edit something in file and then pull)
git pull origin master
(creating a new branch and pushing content of master in new_branch(feature))
git push origin master:feature
git push origin --delete feature

10. Pull Ubuntu image in Docker and execute bash

Follow same steps in Q1. then perform following commands

Sudo yum update -y
Sudo yum install docker
Sudo docker –version
Service docker start
Service docker status
docker images
docker pull ubuntu
docker run -it -d ubuntu
docker ps
docker exec -it (ID in ps) bash
ls
mkdir demo
ls
exit

11. Pull files from Github using git, make changes and push file to Github using Git

(Make a github repo -> create file -> enter some text (ex: “hello world”))

Then open a folder and perform following commands

git init

git pull https://github.com/saishrane-11/demorepo.git

git status

(now make changes in the file in your system)

git add .

git commit -m “new changes done”

git remote add origin https://github.com/saishrane-11/demorepo.git

git push origin master

12. **Pull Docker images, list the Docker images currently stored on your machine, remove an image that you no longer need**

    Follow same steps in Q1. then perform following commands

    Sudo yum update -y
    Sudo yum install docker
    Sudo docker –version
    Service docker start
    Service docker status
    docker images
    docker pull hello-world
    docker images
    docker image rm hello-world
    docker images

13. **Execute the basic command in docker**

    Follow same steps in Q1. then perform following commands

    Sudo yum update -y
    Sudo yum install docker
    Sudo docker –version
    Service docker start
    Service docker status
    docker images
    history

14. **Execute basic command in git**

    git init
    git clone (repo_url)
    git status
    git add .
    git commit -m “first commit”
    git remote add origin (repo_url)
    git push origin master
    git pull (repo_url)
    git checkout -b new_branch
    git checkout master
    git log

15. **Create a Git repository, add a new file to the repository, Commit the changes with a meaningful message, Push the changes to a remote repository.**

    (Same as Q11)

16. **Set up a basic CI pipeline using a CI/CD tool (eg. Jenkins), Configure the pipeline to build a simple project (eg. a “Hello World” application)**

    (Same as Q8)

17. **Build a Docker image form the Docker file, run a Docker container using the built image.**

    Follow same steps in Q1. then perform following commands

    Sudo yum update -y
    Sudo yum install docker
    Sudo docker –version
    Service docker start
    Service docker status
    docker images
    docker pull ubuntu
    docker run -it -d ubuntu
    docker ps