Week 1:

Git & GitHub, JIRA

Git is used for tracking the changes in computer files. It is generally used for source code management in software development. Git is a distributed version control tool it allows multiple developers to work together.

1. Download the Git from the browser and using git we can create some of the git repositories using git commands.
2. Once we have created the repositories, we can push this repository to GitHub account.
3. Some of the git commands like how to create repositories and push to GitHub account.

Commands

1. git init

You can create a repository using the command git init. Navigate to your project folder and enter the command git init to initialize a git repository for your project on the local system.

Syntax: git init

2: git status

Once the directory has been initialized you can check the status of the files, whether they are being tracked by git or not, using the command git status.

Syntax: git status

3: git add

If we want to track all the files in the project folder, we can type the command,

Syntax: git add .

4: git commit

Once the files or changes have been staged, we are ready to commit them in our repository. We can commit the files using the command “git commit –m “custom message”.

Syntax: git commit –m “custom message”

5: git remote

Once everything is ready on our local, we can start pushing our changes to the remote repository. Copy your repository link and paste it in the command git remote add origin “ ”

Syntax: git remote add origin “ https://github.com/maheshkumaravula/demo.git”

6: git push

To push the changes to your repository, enter the command git push origin and hit enter. In our case the branch is master, hence git push origin master. This command will then prompt for username and password, enter the values and hit enter.

Syntax: git push origin <branch name>

7: git clone

If we want to download the remote repository to our local system, we can use the command

Syntax: git clone< https://github.com/maheshkumaravula/demo.git >

9: git branch

Until now, we saw how you can work on git. But now imagine, multiple developers working on the same project or repository. To handle the workspace of multiple developers, we use branches. To create a branch from an existing branch, we type

Syntax: git branch<new branch name>

10: git checkout

To switch to the new branch, we type the command git checkout

Syntax: Git checkout<branch name>

JIRA

Jira is a Bug tracking tool and issue tracking tool it is a project management tool and manual testing tool jira is made by “Atlassian”

1. Here we will be working on issues and product backlogs, how to create product backlogs using agile.
2. We will learn about sprints and how to create sprints and what is epic and bug, story, task, subtask.
3. Sprint planning, daily scrum, sprint review, sprint retro
4. Like sprint backlog and how to solve the issues in sprints
5. We will create a project and we will create an issue for that project and we create the product backlog and then sprint backlogs.

Week 2:

Maven

Maven is a build tool used to manage the entire life cycle of the project, generate reports, and store documents with pom repositories. It is lot more automated and standardized it is mostly used in java-based applications.

1. Launch the instance from cloud and download the Maven.
2. First, we need to install java then we need to install maven by default java is installed on our system

Maven Commands

1. To create a file, we need to type

mvn archetype:generate

Once you have given that command it will ask for

id: 'Enter the default value'

group number:'Enter the default value8'

groupid :'general'

artifactId : 'sample'

property version: '0.1.0'

property: 'Enter the default value'

1. Noe we can check the tree if you do not find the tree install the tree

Sudo apt install tree

3. once you have done this you have to compile

mvn compile

4. Next you need to test the file

mvn test

5. Install the packages

mvn packege

6. At lat you will get the .jar file that is the output of the file.

AZURE and AWS

1. Understanding of AWS and AZURE Clouds. Azure DevOpsgoals.

2. Using cloud instances connected to a ssh.

3. Learned about different cloud instances and connect with the ssh using pem file

Week 3:

Jfrog, Jenkins, Azure Pipelines

Jfrog:

Jfrog antifactory is a tool used in DevOps methodology for multiple purposes. One of the main purposes is to store artifacts that have been created in the code pipeline. Another one is it act as a sort of a buffer for downloading dependencies for the build tools.

1. If we want jfrog we need to install a maven first and then we can install jfrog.
2. Connect to a cloud instance and by using cloud instance we can install jfrog.
3. Go to JFrog on browser

connect maven to jfrog

click on 'libs-sanpshot-local'

enter the 'password'

click on 'generate maven settings'

Use cases:

1. System stability and reliability artifactory and high antifactory

2. security, access control

3. it has support for chef.

Jenkins

Jenkins is open-source automation server written in java. It manages and monitors the changes. It is mostly widely used and adopted and very popular with the open-source community.

1. To connect with Jenkins, we need to create a worker node architecture like one master node and one slave node.
2. We have different commands to create Jenkins's master and slave architecture.
3. Some of the commands are first we need to install java and then from our master system to slave machine.
4. To do that we need to copy the ssh key to slave machine and then it will connect with the master machine.
5. Once we have done that, we can install Jenkins in our master system once we have done the installation, we can copy the ip address of the master machine and paste in the any browser with the ip:8080
6. Now you have successfully connected with Jenkins, there you can create your own projects and node and we can connect with different DevOps tools using Jenkins

AZURE Pipelines

Azure Pipeline is a **cloud service** that we can use to build and test our code project automatically. The Azure pipeline has a lot of capabilities such as continuous integration and continuous delivery to regularly and consistently test and build our code and ship to any target.

1. Go into the Azure DevOps project and click on pipelines. After that, click on the new pipeline button.

2.Now, click on the ***"use the classic editor"*** link down below

3.Select the **project** and **repository** where you want to create the pipeline then click on **Continue**. Click on the **Empty job** link to create a job.

Week 4:

Docker

Docker is a computer programmed that perform os-level virtualization, also known as containerization. It is used to run software packages called containers.

1. First we need to create an instance from cloud and install java and docker once docker is installed we can create an image and containers using docker commands.
2. We can push the image to the docker hub we need to create an docker hub account. And we can push our images to the docker hub using commands.
3. Some of the commands are

1. docker –version

2. docker pull <image name>

3. docker images

4. docker run –it –d <image name>

5. docker ps

6. docker ps –a

7. docker exec –ti <container-id> bash

8. docker stop <container id>

9. docker kill <container id>

10. docker login

11. docker commit <container-id> <image name>

12. docker push <docker hub user name/image name>