**CI&CD**

**INTRODUCTION :** IN general, there is a delay in deploying the s/w based on client requirements to make collaboration and communication b/w dev team and op team we use DevOps

* DevOps practice is CI&CD

**CLOUD DEVOPS:** CICD is applied in cloud environment using AWS / AZURE / GCP

* CI = Merging code changes from multiple developers into a shared repo

TOOL: Jenkins, GitHub Actions

* CD = Automatically preparing the application for release to production

TOOL: AWS PIPELINE, AZURE, GITHUB LAB

**DevOps Lifecycle:**

**Step1:** Dev pushes code into Git repo

**Step2:** Clone the repo into **Jenkins**

**Step3: SonarQube** is used to check quality standards

**Step4:**  Use **Maven** tool to convert the application to **JAR,WAR,EAR**.

**Step5:**  Use **Nexus** repo to save the maven files

**Step6:**  Use **Docker** to convert the app into an image called docker image.

**Step7:**  The runtime of docker image is called Container , and to manage them we use **Kubernetes** (K8S).

**Step8:** To see metrics and logs we use **Prometheus**(Text) and **Grafana**(Graphical)

**VERSION CONTROL SYSTEM :**

* Local VCS
* Centralized VCS
* Distributed VCS (Git with remote)

**LOCAL GIT :** Git is a software used to course code management

* LOCAL GIT = Git installed in the system
* REMOTE GIT : Using service GitHub

**GIT LIFE CYCLE :**

* Git init
* Git add filename
* Git commit -m “msg”
* Git push -u origin {branchname}

**PROCESS :**

* Create a folder and a github repo with the same name
* Open cmd with the path of that folder
* Type gitin which tracks the folders that to be pushed
* Then git add {file\_name} adds the file
* Then git commit -m “msg”
* Create a branch by git branch -M {branchname(main)}
* Add remote repo using cmd :

Git remote add origin {web link}

* Push the file : git push -u origin main