**Jenkins CI/CD**

**CI:** Continuous Integration is a process where we integrate set of process that we follow before delivering application to the customer

**CD:** Continuous Delivery is a process where we deploy application on a specific platform

* Some of the standard steps we follow in cicd process are as follows
* UNIT TESTING
* STATIC CODE ANALYSIS
* CODE QUALITY OR VULNERABILITY TESTING
* AUTOMATION TESTING
* REPORTING
* DEPLOYMENT

**Jenkins:** It is an orchestrator which will integrate all the tools that required for the cicd process

* Jenkins is a java-based application, so if we want to use Jenkins, we need to have java installed as a pre requisite.

**To install java**

Sudo apt update

Sudo apt install openjdk-21-jre-headless

**To install jenkins**

curl -fsSL https://pkg.jenkins.io/debian/jenkins.io-2023.key | sudo tee \

/usr/share/keyrings/jenkins-keyring.asc > /dev/null

echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \

https://pkg.jenkins.io/debian binary/ | sudo tee \

/etc/apt/sources.list.d/jenkins.list > /dev/null

sudo apt-get update

sudo apt-get install jenkins

* By default, Jenkins runs on the port 8080. So after installing Jenkins in the server we need to open it by using **public-ip-address:8080**
* Previously we used to have one ec2 instance installed with Jenkins as a master and the other ec2 instances in which we have Jenkins as the worker, like we used to schedule the things from Jenkins master and the applications work in Jenkins worker. But now we use Jenkins with docker as an agent and as the master, so that no resources will waste at the idle time
* By default, docker runs on a daemon process which doesn’t allow other users to use docker, so for that we need to add the user to the groups for which we wanted docker to be available and must need to restart the machine to add to the group

**To install docker**

Sudo apt install docker.io

**To add docker to the group**

**Sudo su-**

**Usermod –aG docker jenkins**

**Usermod -aG docker ubuntu**

**Systemctl restart docker**

* Once after installing the docker in the machine, we need to install docker pipeline plugin in the Jenkins application
* Log in to Jenkins.
* Go to Manage Jenkins > Manage Plugins.
* In the Available tab, search for "Docker Pipeline".
* Select the plugin and click the Install button.
* Restart Jenkins after the plugin is installed.

**Now we are good to go with writing the first Jenkins pipeline**

* Click on the new item
* Enter the name of the pipeline and click on the pipeline and click on OK

Writing a jenkinsfile in GROOVY script to verify if the docker slave configuration is working as expected

We can use pipeline syntax also get the script to add in groovy file

**Groovy file format**

* **pipeline {**
* **agent {**
* **any**
* **}**
* **stages {**
* **stage(‘1’) {**
* **steps {**
* **echo()**
* **}**
* **}**
* **stage(‘1’) {**
* **steps {**
* **echo()**
* **}**
* **}**
* **}**
* **}**