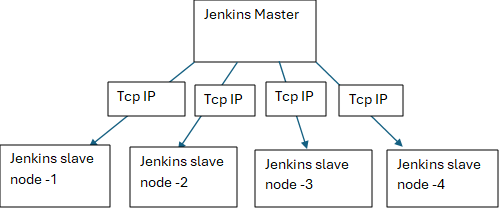
**DevOps: CI/CD with Jenkins pipelines, Maven, Gradle**

**Video-3**

* **Meaning of Continuous Integration:**
* The practice of automatically building a code periodically is called continuous integration.
* **Meaning of Continuous delivery:**
* After building the code can also be deployed are immediately called continuous delivery.
* Code that passes functional and system tests is deployed to production.
* It aims at building, testing, and releasing software faster and more frequently.
* **What are the benefits and why it becomes more popular:**
* Reduced risk
* Increased confidence
* Better quality code
* Ready to ship
* Systematic versioning
* Code quality trend analysis
* Time to market
* Reduced cost
* **Jenkins features:**
* Open source platform for implementing devops pipelines
* Cross platform
* Cost savings
* Plugin ecosystem-Jenkins supports thousands of plugins
* Increased productivity
* **Jenkins Architecture:**
* Jenkins uses the master and slave architecture to manage distributed builds.
* Master node distribute load to slave nodes



* **Jenkins installation:(Using centos)**
* Jenkins official website(wiki.jenkins-ci.org)
* First we need to install java
* Click on installing jenkins on redhat distributions
* Yum install java-1.7.0-openjdk
* To check the version of java and java installed or not (java –version)
* Next step is configuring java home directory and set environment variables:
* Cd/usr/lib/jvm java version path
* Pwd
* U wuill be get the path
* Vi ~/.bash\_profile
* Execute two commands
* Export JAVA\_HOME(paste the java installed path already copied in the above)
* After bin:$JAVA\_HOME/bin
* Set completed java environment variables
* To see the set up properly or not
* Echo $JAVA\_HOME
* Install jenkins:
* sudo wget -O /etc/yum.repos.d/jenkins.repo \  
   https://pkg.jenkins.io/redhat-stable/jenkins.repo
* sudo rpm --import <https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key>
* sudo yum install jenkins
* **Configure jenkins:**
* sudo systemctl start jenkins
* sudo systemctl enable jenkins
* sudo systemctl status Jenkins
* Chkconfig Jenkins on(Jenkins started automatically in any time)
* Ip a(It gives the IP address of server)
* Hostname
* Copy publicip with hostname:8080
* Enter administrator password
* Continue
* Select installed plugins.
* **Get started with Jenkins:**
* Click start using Jenkins
* Jenkins home page will be displayed
* Click on new item
* Enter job name
* Click freestyle project
* Click ok
* **Setup jenkins job:**
* Under build section click execute shell
* Enter script (echo $JAVA\_HOME) (echo $BUILD\_NUMBER)
* Apply and save.
* Click on buildnow.
* **Jenkins pipelines:**
* Jenkins pipeline a speedup plugins which report implementing and integrating continuous delivery pipelines in the jenkins.
* Pipeline provides an extensive setup of tools for modeling simple to complex delivery pipelines.

* **Jenkins pipeline plugins:**
* Go to Jenkins
* Manage Jenkins
* Plugins
* Available
* In search bar enter pipeline plugin (build pipeline plugin)
* Install without restart
* Save.
* **Jenkins pipeline view:**
* Click “+” icon (created job above left side and after all)
* Enter view name
* Select pipeline view
* Click ok
* Enter build pipeline view Title (Title 1)
* Under select initial job (enter job name-for example-test)
* Apply ok
* We need to edit the job
* Click configure and modify.
* **Install artifactory:**
* Artifactory is an open-source repository manager which is published and maintained by company called Jfrog.
* Prerequisites:
* Java
* Go to browser and install java
* To check the version of java and java installed or not (java –version)
* Next step is configuring java home directory and set environment variables:
* Cd/usr/lib/jvm java version path
* Pwd
* U wuill be get the path
* Vi ~/.bash\_profile
* Execute two commands
* Export JAVA\_HOME(paste the java installed path already copied in the above)
* After bin:$JAVA\_HOME/bin
* Set completed java environment variables
* To see the set up properly or not
* Echo $JAVA\_HOME
* Install Jfrog
* Go to browser
* Enter artifactory installation
* Click open source (OSS)
* Copy 3 links and paste in Linux server
* Too start the artifactory service (service artifactory start)
* Significance:
* It means after installation of artifactory you have full installation of Apache tomcat on your server. Install all tomcat webapps also.
* Sudo yum install tomcat-webapps tomcat-admin-webapps
* **BitBucket:**
* It is a remote and free service for source code management.
* Bitbucket is a web-based source control hosting service.
* The projects the use git revision control system.
* Bitbucket offers both commercial plans and free plans.
* **Gradle:**
* For build system we will use Gradle.
* Gradle is an open-source build automation system
* **Jenkins job for java API project**
* **Build CI/CD pipeline: process of integrating Jenkins with bitbucket**
* After seeing source code in bitbucket lets do build CI/CD pipeline for this we can go Jenkins
* Go to Jenkins
* Click on new item
* Enter project name
* Select freestyle project
* Click ok
* Click discard old builds
* Days to keep builds (4)
* Max # of builds to keep (6)
* Under source code management
* Select GIT
* Give project repository URL
* **Note**: If our project repository is public, we didn't provide credentials of GitHub because everyone can access our project and if our project repository is private, we need to give credentials. So that’s why click **add-enter username and password of GitHub-add.**
* Apply
* Save**.**
* **Configure Artifactory:**
* Artifactory is a special repository based on Apache tomcat that can easily store all your artifacts**.**
* The end of the file is in JAR/WAR. This WAR/JAR file could be executable or deployable file, and this file is called artifact.
* our artifact repository manager can easily manage multiple versions of this file.
* **Configure artifactory plugin:**

Go to Jenkins

Click manage Jenkins

Click manage plugins

Available

Serach plugin (artifactory plugin)-this plugin allows deploying maven artifacts and build info to artifactory)

**Configure:**

Click on manage Jenkins

Click configure system

Under artifactory

Select add (give server ID)

Give URL (artifactory installed URL)

Give default deployer credentials (username and password)

Click test connection

Apply

* **Continue building CI/CD pipeline:**
* Go to Jenkins
* Click on job
* Click configure
* Under build environment click Gradle artifactory integration
* Refresh
* Click ext-release-local
* Click project uses the artifactory Gradle plugin
* Click public artifacts to artifactory
* Under build section-select invoke Gradle script
* **Note:** If Gradle is not installed in Jenkins instance
* Click manage Jenkins
* Click global tool configuration
* Under Gradle
* Install automatically.
* Click on job
* Go to configure
* Gradle version- Gradle
* use Gradle wrapper: Gradle wrapper allows Jenkins to download the necessary files and executables from the internet exactly as needed by the Jenkins build job.
* Because any time any Jenkins job need Gradle this wrapper will take care about the installation and configuration of Gradle.
* Click make gradleW executable
* Apply
* Save
* Click build now
* Click console output.
* **CI/CD pipeline with Jenkins and Maven:**
* **Simple java application:**
* Maven: maven is build automation tool used primarily for java projects.
* Maven addresses to aspects for building software

1. Build lifecycle
2. Dependency management

* After completion of building application deploy to tomcat server.
* Open bitbucket
* One file visible there (pom.xml)
* Pom.xml file- the pom.xml file, short for project object model, is an XML file that contains metadata about a maven project. It describes the project’s structure, dependencies, and build processes.
* **Build CI/CD pipeline with Jenkins and maven:**
* Go to Jenkins
* Click new item
* Enter job name(git\_source)
* Select freestyle project
* Ok
* Click git
* Give repository URL
* Under build environment- select delete workspace before build starts
* Under build section-select execute shell-mvn package
* **Note:** If Maven is not installed in Jenkins instance
* Click manage Jenkins
* Click global tool configuration
* Under maven
* Install automatically.
* Under post build actions –select archive the artifacts
* Apply
* Save
* Build now
* Console output
* **Create pipeline view:**
* Click on “+” icon above job
* Enter view name
* Build pipeline view option
* Ok
* Apply
* Save
* Pipeline created with one job.
* We needcreate another job
* Go to new item
* Enter item name
* Select freestyle project
* Ok
* Under build environment-select delete workspace before build starts
* **Note: Copy** artifacts from another job-we need to install copy artifact plugin
* Apply
* Save
* Go to manage Jenkins
* Manage plugins
* Available
* In search bar enter plugin name-copy artifact plugin
* Install without restart
* Go back to Jenkins home page
* Select project
* Configure
* Under build sections-select copy artifacts from another project
* Enter project name (project name is from which project to copy artifacts)
* Which build-latest successful build
* Apply
* Save
* **Note:** Deployment of war file also need one plugin
* Go to manage jenkins
* Manage plugins
* Available
* In search bar enter plugin name-deploy to container plugin
* Install without restart.
* Go back to the top page
* Select project
* Configure
* Under post build actions-select deploy war/ear to a container
* War/jar files-\*\*/\*(deploy everything)\*\*/example.war(if you need particular file)
* Context path-tomcat URL
* Add container(tomcat 7x)
* Provide username and passowrd of tomcat server
* Ener apache tomcat URL
* Apply
* Save.
* **Chain projects in pipeline:**
* Click configure
* Under build triggers section-select build after other projects are built
* Projects to watch(give project name first job name)
* or
* Come to dashboard-select first job-select configure-under post build actions-select build other projects-give second job name
* Apply
* Save
* Click console output (to see output)
* Go to tomcat
* Select manager app
* To see the project name.
* **Manage relational database schema Jenkins and sqitch:**
* Introduction to sqitch:
* Sqitch is a database change management framework which can help us manage database changes in a consistent manner.
* Download sqitch(http://sqitch.org/)
* **Build database schema deployment pipeline with Jenkins and sqitch:**
* Go to Jenkins
* Click new item
* Enter job name(db\_source)
* Click freestyle project
* Click ok
* Under source code management-select git-paste git repository URL
* Under build environment-select delete workspace before build starts
* Under build triggers-select POLL SCM (H/5 \*\*\*\*)
* Under build post actions-select execute shell (cd $WORKSPACE)(add sqitch command)
* Go to manage jenkins-manage pugins-available-build pipeline plugin-install without restart-go back to jenkins home page
* Click “+” icon
* Enter view name
* Select build pipeline view
* Ok
* Apply
* Save
* Its showing no jobs
* Click on new item
* Enter job name
* Select freestyle project
* Ok
* Under SCM give git repo url
* Under post build actions-select execute shell-give commands(cd $WORKSPACE)(add sqitch command)
* Apply
* Save
* Configure
* select build other projects
* Apply
* Save
* **Operational considerations for Jenkins:**
* Securing Jenkins:
* Click manage Jenkins
* Click configure global security
* JNLP (connect slave nodes to master)
* Select random
* access control-Jenkins own user database
* Authorization-matrix based security-root (give all permissions)
* Apply
* Save
* Click on manage Jenkins
* Click on manage users
* create user(enter username, password, confirm password, full name and email)
* Click on create user
* Click manage Jenkins
* Global security
* add created user
* give permissions
* Apply save
* **Scaling Jenkins:**
* Jenkins use master slave architecture
* The main aim is distributing build workloads across multiple slave nodes, reducing the load on the master node.