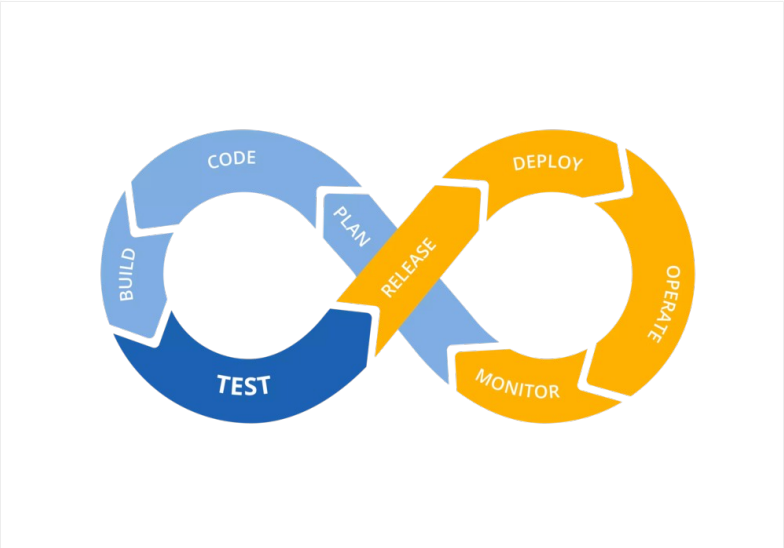


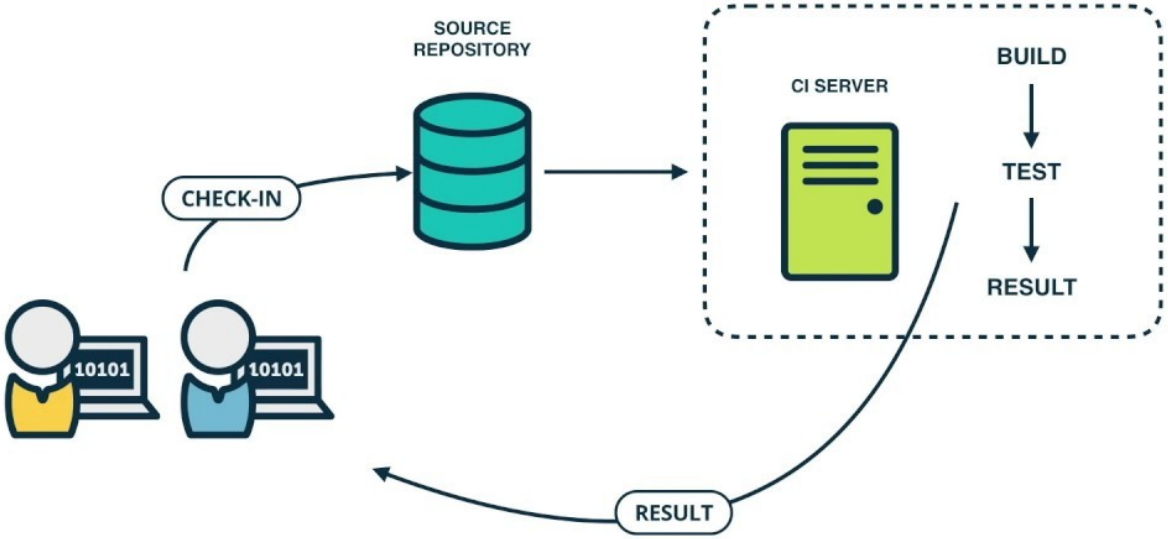
Continuous Integration and Delivery (CI/CD)

WHAT IS CI/CD ?

The practice of automating the integration of code changes from multiple contributors into a single software project

Continuous integration leads to continuous deployment allowing us to deliver software more rapidly

Continuous Integration (CI)



Jenkins

Open Source Continuous Integration Server

WHAT IS JENKINS?

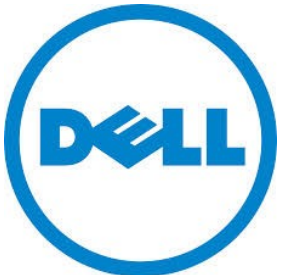
Monitors executions of repeated jobs

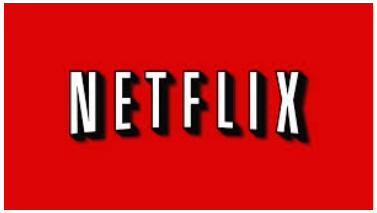
Used primarily for continuously building and testing software projects, i.e. “Continuous Integration” or “CI”

Previously known as “Hudson”

http://jenkins-ci.org

WHO USES JENKINS?





WHY USE JENKINS?

Immediate feedback on broken builds

No building from dev machines and manually copying files

Automated deployment

Automated test execution and feedback

PLUG-INS

Over 600 plug-ins available for customization

Examples:

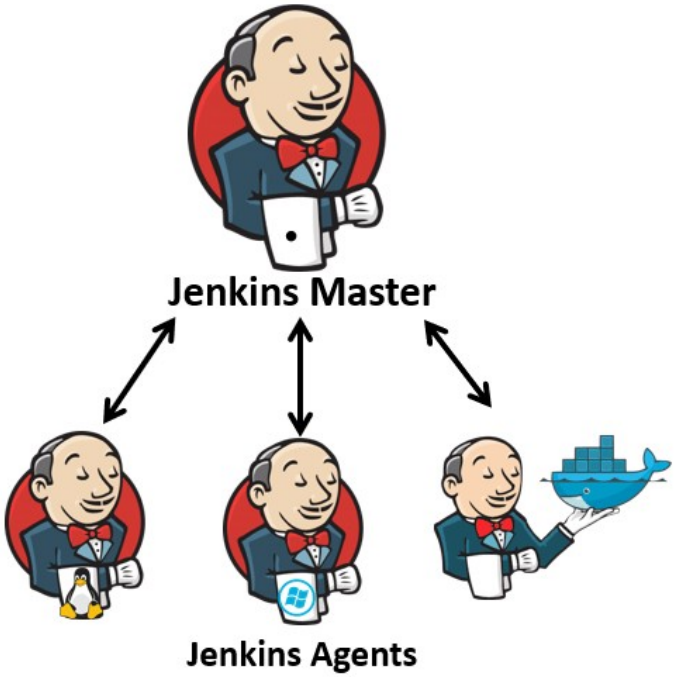
- JIRA
- Ant
- Maven
- JUnit
- Javadoc
- SSH
- MSBuild
- NUnit
- PowerShell
- Email Notifications

https://wiki.jenkins-ci.org/display/JENKINS/Plugins

DISTRIBUTED BUILDS

Jenkins architecture is fundamentally "Master+Agent“

The master is designed to do co-ordination and provide the GUI and API endpoints, and the Agents are designed to perform the work



QUESTIONS?

INSTALLATION

Prerequisites :

Create a VM with Vmware/Virtualbox using image centos

Logged to your machine as root (or user with sudo privileges)

Installing Jenkins :

1. Java is a java application, so you need to install java

```
$ sudo yum install java-1.8.0-openjdk-devel
```

2. Enable Jenkins repository

```
$ curl --silent --location
http://pkg.jenkins-ci.org/redhat-stable/jenkins.repo | sudo
tee /etc/yum.repos.d/jenkins.repo
```

```
$ sudo rpm --import https://jenkins-ci.org/redhat/jenkins
ci.org.key
```

INSTALLATION

3. Install and start Jenkins

```
$ sudo yum install jenkins
$ sudo systemctl start jenkins
$ sudo systemctl status jenkins
$ sudo systemctl enable jenkins
```

4. Allow Jenkins port to be accessible from outsite the VM

```
$ sudo firewall-cmd --permanent --zone=public --add-port=8080/tcp
$ sudo firewall-cmd –reload
```

5. Open http://your_ip_or_domain:8080 and setup your jenkins

You can find more at link below :

JENKINS PIPELINE

A continuous delivery pipeline is an automated expression of your process for getting software from version control right through to your users and customers

Jenkinsfile (Declarative Pipeline)

```
pipeline {
    agent any
    stages {
        stage('Build') {
            steps {
                sh 'makemvn -B -DskipTests clean package'
            }
        }
        stage('Test'){
            steps {
                sh  'mvn test'
            }
            post {
               always {
                    junit 'target/surefire-reports/*.xml'
                }
            }
        }
        stage('Deliver') {
            steps {
                sh './jenkins/scripts/deliver.sh'
            }
        }
    }
}
```

QUESTIONS?

EXAMPLE 1

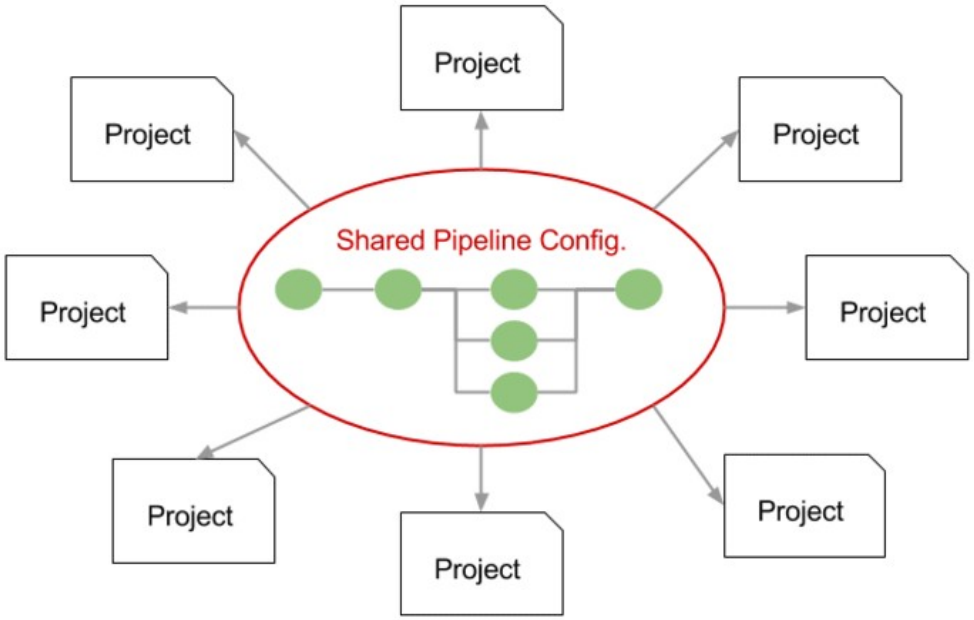
Build a java app with maven

1. Clone https://github.com/ericsonrumuy7/Jenkins-future.git to your local machine
2. Open your jenkins and create pipeline job
3. Setup post-commit git to trigger Jenkins pipeline job

More info :

CENTRALIZED PIPELINE

As Pipeline is adopted for more and more projects in an organization, common patterns are likely to emerge. Oftentimes it is useful to share parts of Pipelines between various projects to reduce redundancies and keep code "DRY"



EXAMPLE 2

Implement Jenkins shared library

1. Clone https://github.com/ericsonrumuy7/Jenkins-future.git to your local machine
2. Configure global library pipeline from Jenkins UI
3. Update your Jenkinsfile in maven repo (example 1) to use library

More info :

QUESTIONS?

TAKS 1

Deploy java maven application (example 1) to targeted VM using shared library

TAKS 2

Deploy using agent Jenkins

QUESTIONS?