DEVOPS – CI/CD JENKINS

23-11-2020 to 27-11-2020

Venkatesh Reddy Madduri && ShankarPrasad

Agenda

- ✓ Introduction
- ✓ Git & GitHub
- ➢ **Continuous Integration Using Jenkins**
- ➢ Introduction to Docker
- ➢ Kubernetes
- ➢ Ansible
- ➢ Continuous Monitoring using Nagios
- ➢ DevOps Project

What is Jenkins

- Jenkins is an open source automation tool written in Java programming language that allows continuous integration.

- Jenkins builds and tests our software projects which continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build.

- It also allows us to continuously deliver our software by integrating with a large number of testing and deployment technologies.

- Jenkins offers a straightforward way to set up a continuous integration or continuous delivery environment for almost any combination of languages and source code repositories using pipelines, as well as automating other routine development tasks.

What is Jenkins ..

With the help of Jenkins, organizations can speed up the software development process through automation. Jenkins adds development life-cycle processes of all kinds, including build, document, test, package, stage, deploy static analysis and much more.

Jenkins achieves CI (Continuous Integration) with the help of plugins. Plugins is used to allow the integration of various DevOps stages. If you want to integrate a particular tool, you have to install the plugins for that tool. For example: Maven 2 Project, Git, HTML Publisher, Amazon EC2, etc.

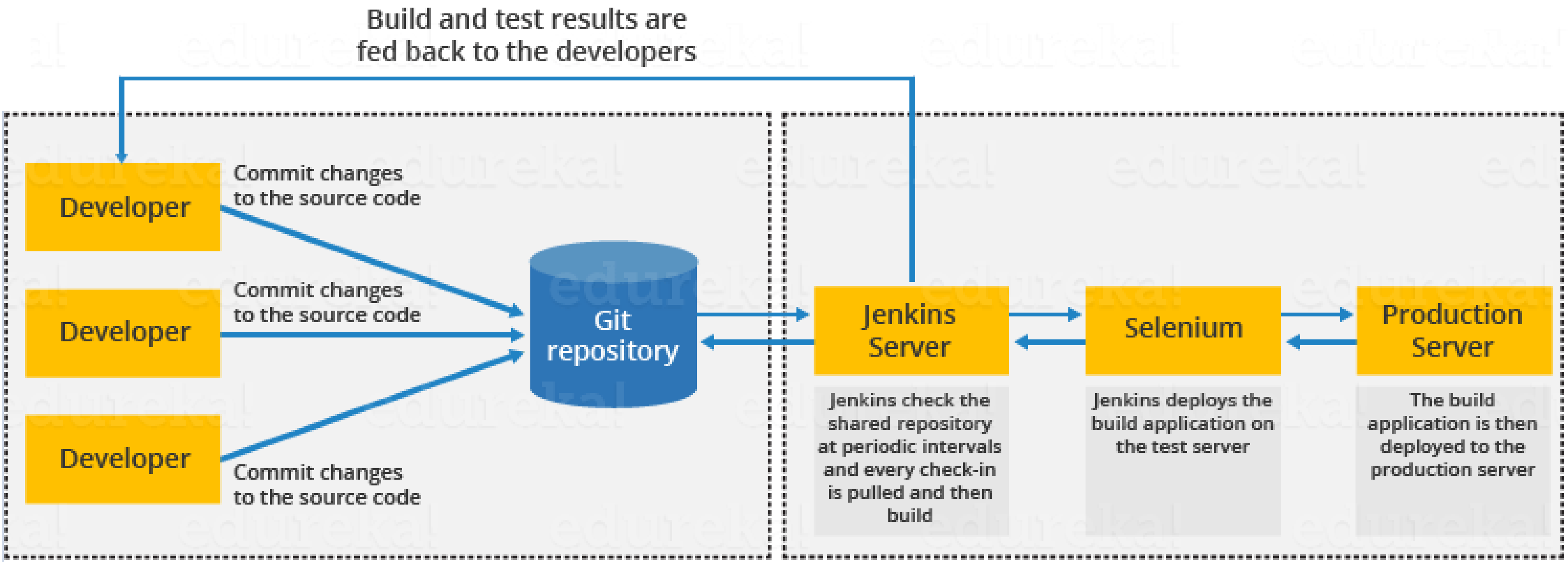
For example: If any organization is developing a project, then Jenkins will continuously test your project builds and show you the errors in early stages of your development.

Possible steps executed by Jenkins are for example:

Perform a software build using a build system like Gradle or Maven Apache

- Execute a shell script.
- Archive a build result.
- Running software tests.

Jenkins Work Flow


Build and test results are fed back to the developers
Developer
Commit changes to the source code
Developer
Commit changes to the source code
Developer
Commit changes to the source code
Git repository
Jenkins Server
Selenium
Production Server
Jenkins check the shared repository at periodic intervals and every check-in is pulled and then build
Jenkins deploys the build application on the test server
The build application is then deployed to the production server

Advantages and Disadvantages of Jenkins

Advantages of Jenkins

- It is an open source, free of cost, Easy support since widely used.
- It does not require additional installations or components. Means it is easy to install.
- Easily configurable.
- It supports 1000 or more plugins to ease your work. If a plugin does not exist, you can write the script for it and share with community.
- It is built in java and hence it is portable.
- It is platform independent. It is available for all platforms and different operating systems. Like OS X, Windows or Linux.
- Jenkins also supports cloud based architecture so that we can deploy Jenkins in cloud based platforms.

Advantages and Disadvantages of Jenkins

Disadvantages of Jenkins

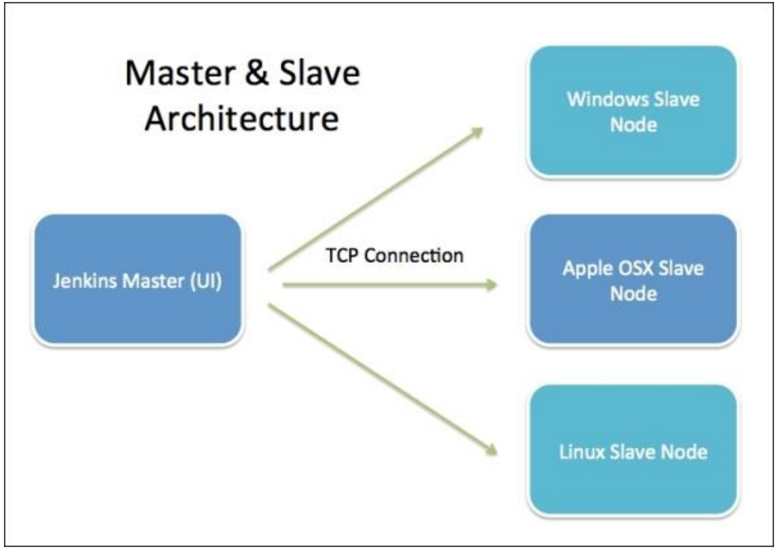
- Its interface is out dated and not user friendly compared to current user interface trends.
- Not easy to maintain it because it runs on a server and requires some skills as server administrator to monitor its activity.
- CI regularly breaks due to some small setting changes. CI will be paused and therefore requires some developer's team attention.

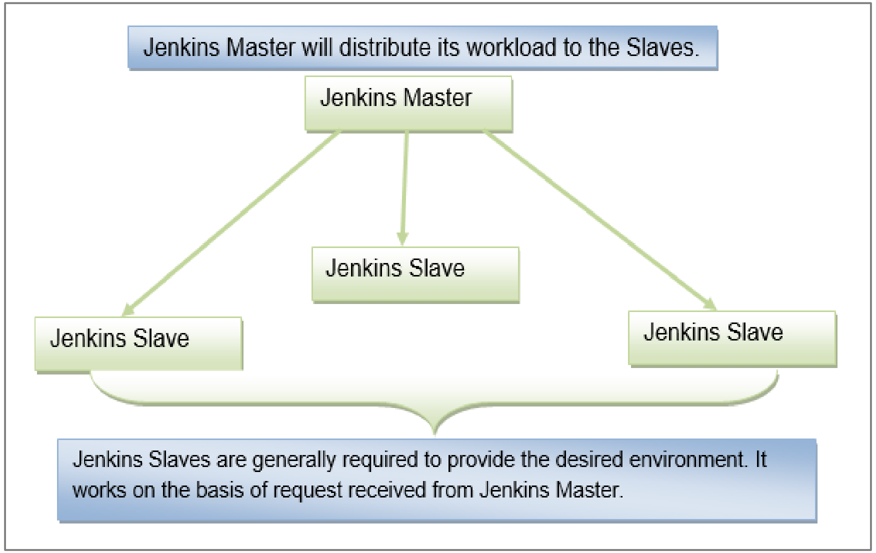
Jenkins Architecture

Jenkins follows Master-Slave architecture to manage distributed builds. In this architecture, slave and master communicate through TCP/IP protocol.

Jenkins architecture has two components:

- ➢ Jenkins Master/Server
- ➢ Jenkins Slave/Node/Build Server





Jenkins Master will distribute its workload to the Slaves.

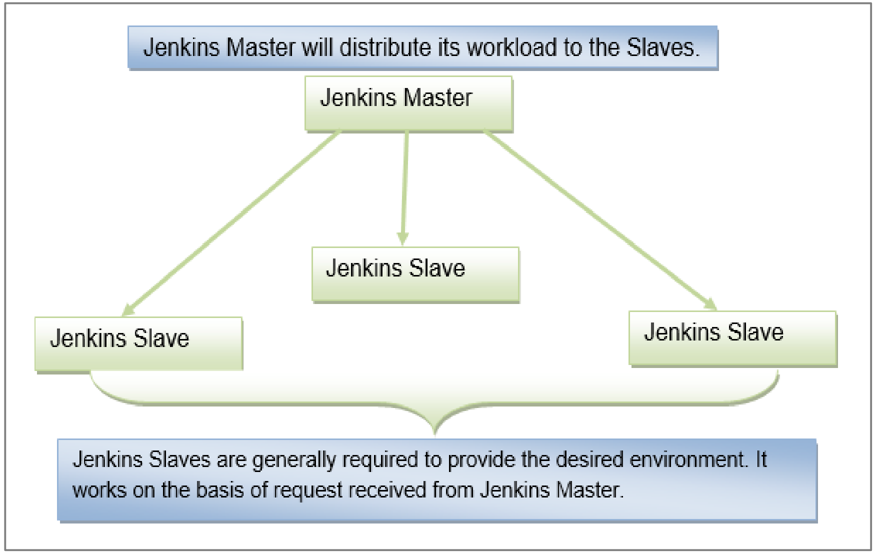
Jenkins Slaves are generally required to provide the desired environment. It works on the basis of request received from Jenkins Master.

Jenkins Master

- The main server of Jenkins is the Jenkins Master. It is a web dashboard which is nothing but powered from a war file. By default it runs on 8080 port. With the help of Dashboard, we can configure the jobs/projects but the build takes place in Nodes/Slave. By default one node (slave) is configured and running in Jenkins server. We can add more nodes using IP address, user name and password using the ssh, jnlp or webstart methods.
- The server's job or master's job is to handle:
- Scheduling build jobs.
- Dispatching builds to the nodes/slaves for the actual execution.
- Monitor the nodes/slaves (possibly taking them online and offline as required).
- Recording and presenting the build results.
- A Master/Server instance of Jenkins can also execute build jobs directly.

Jenkins Slave

- Jenkins slave is used to execute the build jobs dispatched by the master. We can configure a project to always run on a particular slave machine, or particular type of slave machine, or simple let the Jenkins to pick the next available slave/node.
- As we know Jenkins is developed using Java is platform independent thus Jenkins Master/Servers and Slave/nodes can be configured in any servers including Linux, Windows, and Mac.



Jenkins Master will distribute its workload to the Slaves.

Jenkins Slaves are generally required to provide the desired environment. It works on the basis of request received from Jenkins Master.

Installing Jenkins

Hardware Requirements	
Memory	2 GB RAM (Recommended)
Disk Space	We need at least 1 GB of space in our hard drive for Jenkins

Software Requirements	
JDK	We need either Java Development (JDK) or Java Runtime Environment (JRE)
Operating System	Jenkins can be installed on Windows, Mac OS X, Ubuntu/Debian, Red Hat/Fedora/CentOS, openSUSE, FReeBSD, OpenBSD, Gentoo.
Java Container	The WAR (Web Application Resource) file can be run in any container that supports Servlet 2.4/JSP 2.0 or later. (For example Tomcat 5).

Release Types

Long term support release (LTS): Long-term support releases (LTS) are available every 12 weeks. They are stable and are widely used and tested. Basically, this release is intended for end users.

Weekly release: This release is available in every week by fixing bugs in its earlier version. These releases are intended towards plugin developers.

Jenkins Setup

- Update /etc/yum.repos.d/jenkins.repo

```
#sudo wget -O /etc/yum.repos.d/jenkins.repo
https://pkg.jenkins.io/redhat-stable/jenkins.repo
```

- Import jenkins.io.key

```
#sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key
```

- Install any latest upgrade

```
#sudo yum upgrade
```

- Install Jenkins and Java

```
#sudo yum install jenkins java-1.8.0-openjdk-devel
```

- Reload the configuration

```
#sudo systemctl daemon-reload
```

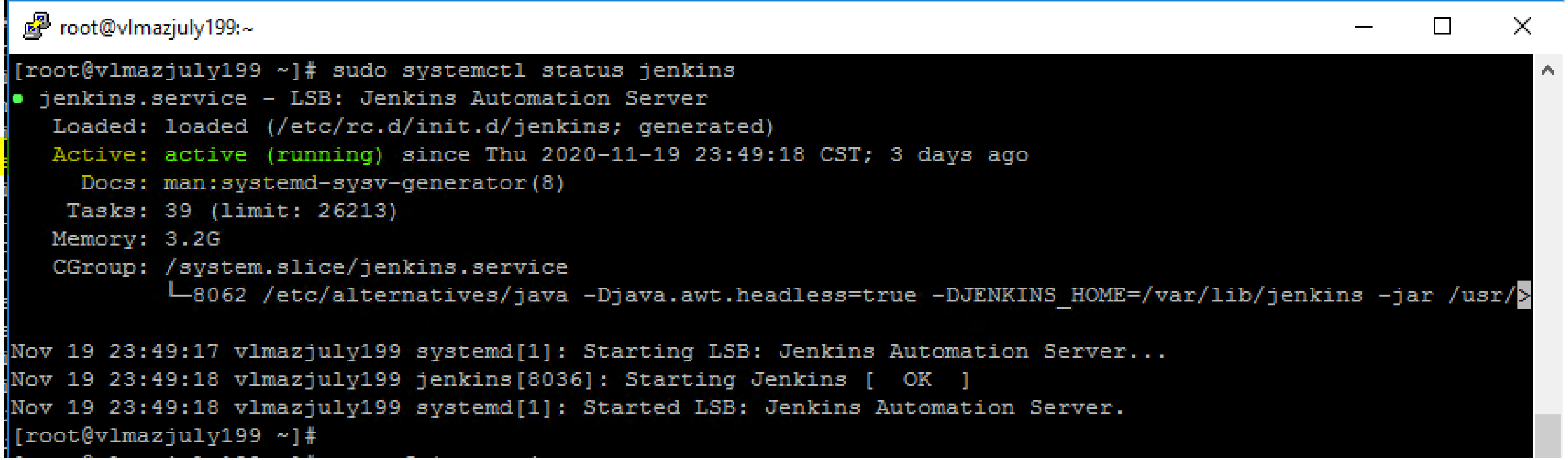
- Start, Check Status and Stop Jenkins

```
#sudo systemctl start jenkins

#sudo systemctl status jenkins

#sudo systemctl stop jenkins
```

Jenkins Setup Verification

sudo systemctl status jenkins

Note :-Type ‘q’ for command prompt after above command.

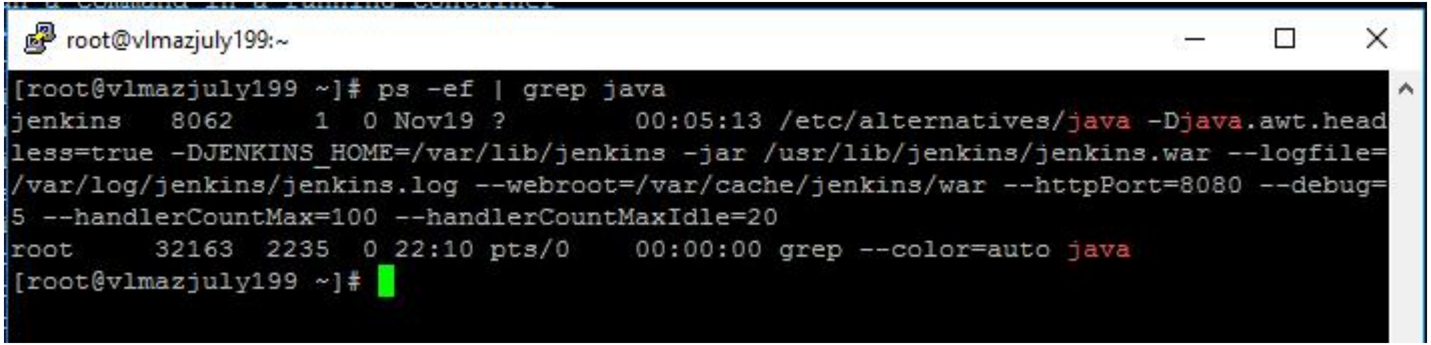


```
[root@vlmazjuly199 ~]# sudo systemctl status jenkins
● jenkins.service - LSB: Jenkins Automation Server
   Loaded: loaded (/etc/rc.d/init.d/jenkins; generated)
   Active: active (running) since Thu 2020-11-19 23:49:18 CST; 3 days ago
     Docs: man:systemd-sysv-generator(8)
    Tasks: 39 (limit: 26213)
   Memory: 3.2G
   CGroup: /system.slice/jenkins.service
           └─8062 /etc/alternatives/java -Djava.awt.headless=true -DJENKINS_HOME=/var/lib/jenkins -jar /usr/>

Nov 19 23:49:17 vlmazjuly199 systemd[1]: Starting LSB: Jenkins Automation Server...
Nov 19 23:49:18 vlmazjuly199 jenkins[8036]: Starting Jenkins [  OK  ]
Nov 19 23:49:18 vlmazjuly199 systemd[1]: Started LSB: Jenkins Automation Server.
[root@vlmazjuly199 ~]#
```

Jenkins Setup Verification

ps –ef | grep java

Note :-Check Jenkin Java Process is running.



```
[root@vlmazjuly199 ~]# ps -ef | grep java
jenkins   8062     1  0 Nov19 ?        00:05:13 /etc/alternatives/java -Djava.awt.head
less=true -DJENKINS_HOME=/var/lib/jenkins -jar /usr/lib/jenkins/jenkins.war --logfile=
/var/log/jenkins/jenkins.log --webroot=/var/cache/jenkins/war --httpPort=8080 --debug=
5 --handlerCountMax=100 --handlerCountMaxIdle=20
root     32163  2235  0 22:10 pts/0    00:00:00 grep --color=auto java
[root@vlmazjuly199 ~]#
```

Thank you