

Name – Sanjay Raghuwanshi

Email – raghu1c.sanjay@gmail.com

Architecting Jenkins Pipeline for Scale.

Project 2

DESCRIPTION

Use Jenkins to set up a distributed pipeline that will compile and test a Maven project on two different slave nodes respectively.

Background of the problem statement:

You're a DevOps engineer at Softmax Solutions, a software company that develops image filters for various photo enhancement apps. It is undergoing an infrastructural change to implement DevOps in its development process. The company uses git as their Source Code Management System and AWS for hosting its servers. You're required to architect a scalable Jenkins Pipeline for building and testing the software stack. You're tasked with designing a Jenkins architecture that involves one master and two slave nodes, all hosted on various AWS instances. The build jobs should always be triggered by the master and executed on the slaves. You have to set up a pipeline on the master node and write a Groovy script that clearly differentiates the tasks to be run on various slaves.

You must use the following:

- **Git:** As a version control system for the software
- **Jenkins:** To create the build pipeline
- **Spring boot:** To create the Maven app
- **Maven:** To compile the program
- **AWS EC2:** To run the master and slave nodes

The following requirements should be met:

- The app should be built with Maven.
- There should be three EC2 instances to run the master and two slave nodes.
- All builds should be triggered and monitored by the master node.
- Compilation and testing should be done on dedicated slave nodes.

1. Created 3 EC2 instances on AWS -

Instances (3) Info										
<input type="text" value="Filter instances"/>										
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
<input type="checkbox"/>	Master	i-08913f9d6a2e6ab64	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c	ec2-3-235-136-126.co...	3.235.136.126	-
<input type="checkbox"/>	Slave1	i-00c6b8b7fe093d7f4	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c	ec2-3-91-172-41.comp...	3.91.172.41	-
<input type="checkbox"/>	Slave2	i-089c76ac05a7a15f4	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c	ec2-3-231-29-58.comp...	3.231.29.58	-

2. Start jenkins on master -

```
ubuntu@ip-172-31-10-44:~$
ubuntu@ip-172-31-10-44:~$ service jenkins status
● jenkins.service - LSB: Start Jenkins at boot time
   Loaded: loaded (/etc/init.d/jenkins; generated)
   Active: active (exited) since Thu 2021-07-01 09:57:27 UTC; 2min 32s ago
     Docs: man:systemd-sysv-generator(8)
    Tasks: 0 (limit: 1160)
   Memory: 0B
    CGroup: /system.slice/jenkins.service

Jul 01 09:57:25 ip-172-31-10-44 systemd[1]: Starting LSB: Start Jenkins at boot time...
Jul 01 09:57:25 ip-172-31-10-44 jenkins[5375]: Correct java version found
Jul 01 09:57:25 ip-172-31-10-44 jenkins[5375]: * Starting Jenkins Automation Server jenkins
Jul 01 09:57:25 ip-172-31-10-44 su[5419]: (to jenkins) root on none
Jul 01 09:57:25 ip-172-31-10-44 su[5419]: pam_unix(su-l:session): session opened for user jenkins by (uid=0)
Jul 01 09:57:26 ip-172-31-10-44 su[5419]: pam_unix(su-l:session): session closed for user jenkins
Jul 01 09:57:27 ip-172-31-10-44 jenkins[5375]: ...done.
Jul 01 09:57:27 ip-172-31-10-44 systemd[1]: Started LSB: Start Jenkins at boot time.
ubuntu@ip-172-31-10-44:~$
```

3. Setting up Jenkins on master node -

Sign in [Jenkins] x +

ec2-3-235-136-126.compute-1.amazonaws.com:8080/login?from=%2F

Getting Started

Unlock Jenkins

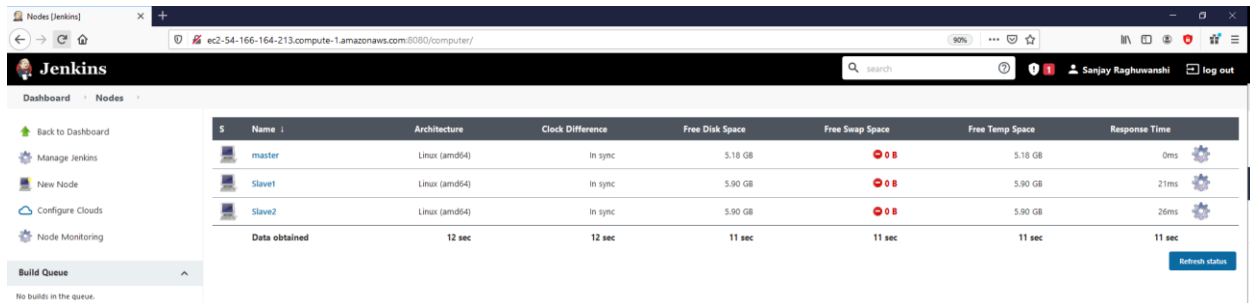
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

`/var/lib/jenkins/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

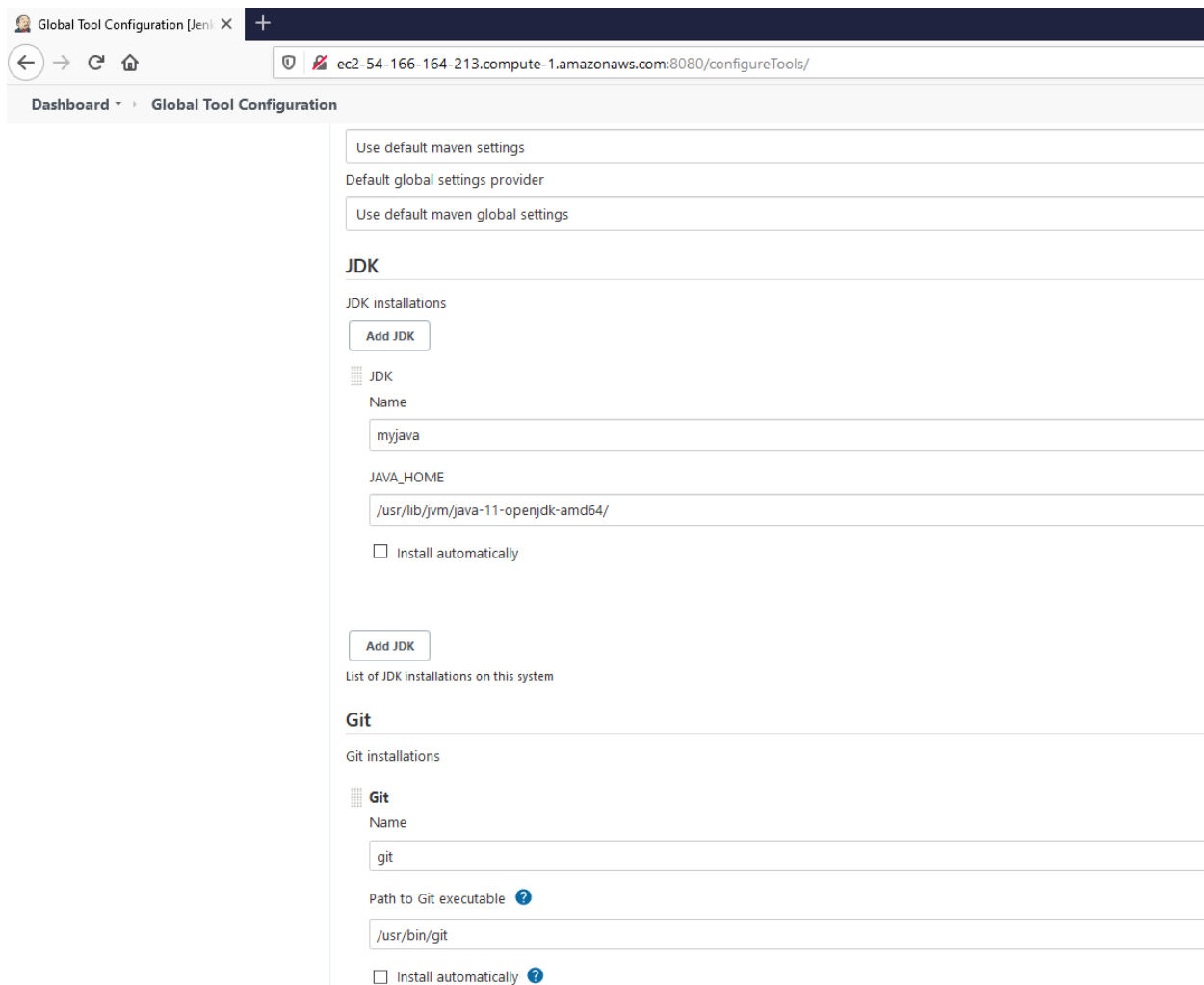
4. Setting up agents –



The screenshot shows the Jenkins 'Nodes' page. On the left is a sidebar with links: 'Back to Dashboard', 'Manage Jenkins', 'New Node', 'Configure Clouds', and 'Node Monitoring'. The main area displays a table of nodes. The table has columns: 'Name', 'Architecture', 'Clock Difference', 'Free Disk Space', 'Free Swap Space', 'Free Temp Space', and 'Response Time'. There are three nodes listed: 'master', 'Slave1', and 'Slave2'. All are Linux (amd64) and 'In sync'. Below the table, a summary row shows 'Data obtained' and various time metrics. A 'Refresh status' button is at the bottom right.

Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
master	Linux (amd64)	In sync	5.18 GB	0 B	5.18 GB	0ms
Slave1	Linux (amd64)	In sync	5.90 GB	0 B	5.90 GB	21ms
Slave2	Linux (amd64)	In sync	5.90 GB	0 B	5.90 GB	26ms
Data obtained		12 sec	12 sec	11 sec	11 sec	11 sec

5. Configure Git, Java & Maven in Global Tool Configuration –



The screenshot shows the 'Global Tool Configuration' page in Jenkins. It has sections for Maven, JDK, and Git. The Maven section has three checkboxes, all of which are selected. The JDK section has an 'Add JDK' button, a table with one entry 'myjava' with its path, and an 'Install automatically' checkbox. The Git section has an 'Add Git' button, a table with one entry 'git' with its path, and an 'Install automatically' checkbox.

Use default maven settings

Default global settings provider

Use default maven global settings

JDK

JDK installations

Add JDK

JDK
myjava

JAVA_HOME

/usr/lib/jvm/java-11-openjdk-amd64/

☐ Install automatically

Add JDK

List of JDK installations on this system

Git

Git installations

Git
git

Path to Git executable

/usr/bin/git

☐ Install automatically

6. Pipeline Syntax –

Pipeline

Definition

Pipeline script

Script

```
1 pipeline {
2   agent none;
3   tools {
4     maven 'my-maven'
5     jdk 'my-java'
6   }
7   stages {
8     stage('Checkout the code') {
9       agent {node 'master'}
10      steps {
11        git branch: 'master', url: 'https://github.com/sanjayraghuwanshi/JenkinsProject2.git'
12      }
13    }
14    stage('Code compile') {
15      agent {node 'Slave1'}
16      steps {
17        sh """
18          rm -rf /home/ubuntu/workspace/pipeline-project2/JenkinsProject2
19          git clone https://github.com/sanjayraghuwanshi/JenkinsProject2.git
20          cd /home/ubuntu/workspace/pipeline-project2/JenkinsProject2
21          mvn compile
22          sleep 60
23        """
24      }
25    }
26    stage('JUnit Test') {
27      agent {node 'Slave2'}
28      steps {
29        sh """
30          rm -rf /home/ubuntu/workspace/pipeline-project2/JenkinsProject2
31          git clone https://github.com/sanjayraghuwanshi/JenkinsProject2.git
32          cd /home/ubuntu/workspace/pipeline-project2/JenkinsProject2
33          mvn test
34          sleep 60
35        """
36      }
37    }
38    stage('Packaging') {
39      agent {node 'Slave2'}
40      steps {
41        sh """
42          cd /home/ubuntu/workspace/pipeline-project2/JenkinsProject2
43          mvn package
44        """
45      }
46    }
47  }
48  post {
49    always {
50      node {node 'Slave2'}
51      junit 'JenkinsProject2/target/surefire-reports/**/*.xml'
52    }
53  }
54 }
55 }
```

☒ Use Groovy Sandbox

7. Pipeline view for different stages with test results –

Pipeline pipeline-project2

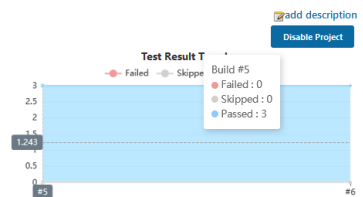
Recent Changes

Stage View


Average stage times:
(Average full run time: ~2min 38s)

Jul 01 17:18 No Changes
Jul 01 17:02 No Changes


Checkout the code	Code Compile	JUnit Test	Packaging	Declarative: Post Actions
649ms	1min 13s	1min 15s	6s	498ms
663ms	1min 14s	1min 15s	6s	133ms
635ms	1min 13s	1min 15s	7s	864ms





8. Demonstration of different agents while executing the pipeline –


**Jenkins**


[Dashboard](#) ▾ ▸


 New Item


 People


 Build History


 Project Relationship

 Check File Fingerprint

 Manage Jenkins


 My Views

 Lockable Resources

 New View


Build Queue ^
No builds in the queue.



Build Executor Status ^


 master

1 Idle

2 Idle

 Slave1

1  pipeline-project2 #4 (Code Compile) 

 Slave2

1 Idle

**Jenkins**[Dashboard](#) ▾ ▸

 New Item

 People

 Build History

 Project Relationship

 Check File Fingerprint

 Manage Jenkins

 My Views

 Lockable Resources

 New View

Build Queue ^
No builds in the queue.**Build Executor Status** ^

 master

1 Idle

2 Idle

 pipeline-project2 #4 

 Slave1

1 Idle

 Slave2

1  pipeline-project2 #4 (JUnit Test) 

**Jenkins**[Dashboard](#) ▾ ▸

 New Item

 People

 Build History

 Project Relationship

 Check File Fingerprint

 Manage Jenkins

 My Views

 Lockable Resources

 New View

Build Queue ^
No builds in the queue.**Build Executor Status** ^

 master

1 Idle

2 Idle

 Slave1

1 Idle

 Slave2

1  pipeline-project2 #4 (Packaging) 