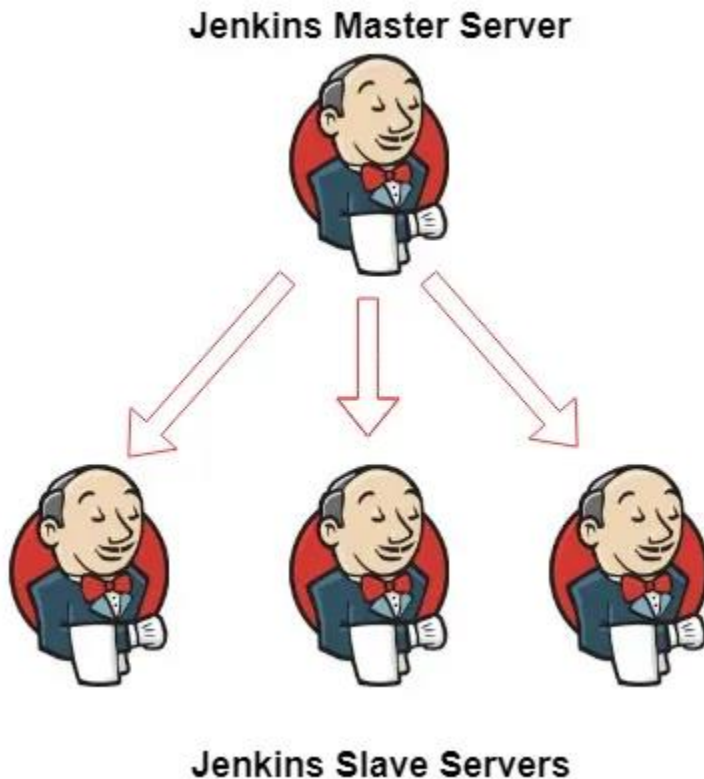


Day 28 Task: Jenkins Agents

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

To run heavy projects which gets build on a regular basis is not a good option.

In such a scenario, need to off load, load from Master by configuring more slaves.



Distributed builds are used to absorb extra load or to run specialized build jobs in a specific operating system or environments.

Jenkins slaves are generally required to provide the desired environment. They work on the basis of requests received from Jenkins master.

Jenkins Master (Server): -

Jenkins's server or master node holds all key configurations. Jenkins master server is like a control server that orchestrates all the workflow defined in the pipelines. For example,

- Scheduling build jobs.

- Dispatching builds to the slaves for the actual execution.
- Monitor the slaves (possibly taking them online and offline as required).
- Recording and presenting the build results.
- A Master instance of Jenkins can also execute build jobs directly.

Jenkins Agent(slave): -

An agent is typically a machine or container that connects to a Jenkins master and this agent that actually execute all the steps mentioned in a Job. When you create a Jenkins job, you have to assign an agent to it. Every agent has a label as a unique identifier.

When you trigger a Jenkins job from the master, the actual execution happens on the agent node that is configured in the job.

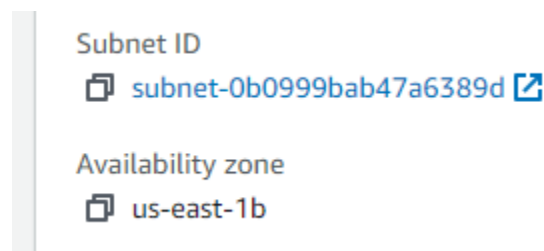
A single, monolithic Jenkins installation can work great for a small team with a relatively small number of projects. As your needs grow, however, it often becomes necessary to scale up. Jenkins provides a way to do this called “master to agent connection.” Instead of serving the Jenkins UI and running build jobs all on a single system, you can provide Jenkins with agents to handle the execution of jobs while the master serves the Jenkins UI and acts as a control node.

Full Jenkins installation on a slave is not required.

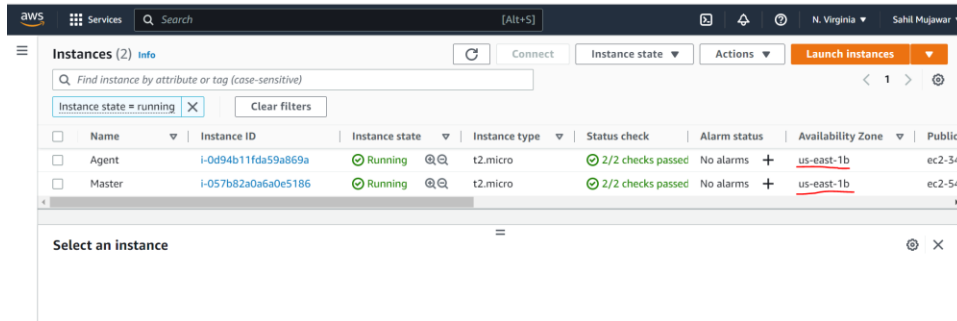
Follow below steps to run a project using Jenkins Agent: -

1] before starting connecting Jenkins agent to Jenkins master, first create a new EC2 Instance, same .as you create master one.

Note: - [Make sure Your Jenkins master SubnetID and Availability one is same as Jenkins slave.]



This SubnetID and Availability zone ,you can find in networking TAB.



2] After Creating an agent EC2 instance, you can make a connection between master and slave.

1. Go to master command prompt and type

ssh-keygen

After that **.ssh** folder is created, this folder contains your public and private key

```
ubuntu@ip-172-31-92-26:~$ cd /home/ubuntu/
ubuntu@ip-172-31-92-26:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:2Tv/bWf1wsEUS3P+td5Hp5hdbwCaQpNSYvN/gaqbDVE ubuntu@ip-172-31-92-26
The key's randomart image is:
+---[RSA 3072]-----+
|
| o=.
| o .o=
| O E ...o
| o X o oo *
| S = . ooB
| o + .O+
| . + =.*
| = o . oo
| +.. ..
+---[SHA256]-----+
ubuntu@ip-172-31-92-26:~$ cd .ssh
ubuntu@ip-172-31-92-26:~/.ssh$ ls
authorized_keys  id_rsa  id_rsa.pub
```

id_rsa -----> private key

id_rsa.pub -----> public key

2. Copy this master public key and paste into agent authorized key.

```
ubuntu@ip-172-31-81-198:~$ cd /home/ubuntu/
ubuntu@ip-172-31-81-198:~$ cd .ssh
ubuntu@ip-172-31-81-198:~/.ssh$ ls
authorized_keys
ubuntu@ip-172-31-81-198:~/.ssh$ vi authorized_keys
ubuntu@ip-172-31-81-198:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDQ2ryao6nt1qD4nZuvvsYM/D3nH5jMg2mY9AfuWp1fbbQcT+TDhAGiNx09nmZRBKCB9GXvncCT944KU1218mLfUUVV
ETm7h2tXp+/OUKz7VD6vu629o70R5ChFkEabdjVqMjsh6he+CQkFrBjIwInt0h/yTPKQpUrMSxcZ0h7f0MA2XLhoEjJ2tFv05ZquI051BZOZOisFPSFLxMgsEQg+u5:
3L43Icu8P1QUMfLgVWjHGDvRIRKegRCi+9 Rushi-new-keypair
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDQ2ryao6nt1qD4nZuvvsYM/D3nH5jMg2mY9AfuWp1fbbQcT+TDhAGiNx09nmZRBKCB9GXvncCT944KU1218mLfUUVV
L83bUdesqH9COA7JKCVQ2kmdYwTbhf+yHq6+6a9woou3jWRWUB2C63SLjVrD0V4YmpJOjZK1I10rcHLzg2kZiFxpixZKjW1SWMqFcQQDqqrFMDL64nSI+niafIc
QAAdO9oQjAl0KmpJlWYR9XzSiV8X7crzGHOGsTghIQPSAFce5bWeyZVpn8cR99NxjQ1hu819QJ58GOCjgYwF40NZFVNf0XqCaPL60mk9pwiJaEvSaGPK6Lit8U0lwGI
VFESrpVv+AKPj36NqYR5eubqBI0BpOdMk= ubuntu@ip-172-31-92-26
```

3. After that go to **Manage Jenkins** and click on **Manage nodes and clouds**

Click on New Node and type mode name

Dashboard > Manage Jenkins > Nodes >

+ New Node

Configure Clouds

Node Monitoring

Build Queue

Build Executor Status

Build-In Node

1 Idle

2 Idle

agent1

1 Idle

New node

Node name

Type

☐ Permanent Agent

Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.

☐ Copy Existing Node

Create

Also add the Labels and inside Launch method select Launch agents via ssh.

Dashboard > Manage Jenkins > Nodes > agent1

System Information

Disconnect

Build Executor Status

1 Idle

Labels

agent1

Usage

Only build jobs with label expressions matching this node

Launch method

Launch agents via SSH

Host

34.238.246.100

Credentials

ubuntu (This is agentkey)

+ Add

Host Key Verification Strategy

Save

Inside Launch Method, put Host Ip address and Jenkins master Private key Using **SSH username and private key**

It looks like below image.

Dashboard > Manage Jenkins > Nodes >

+ New Node

Configure Clouds

Node Monitoring

Build Queue

Build Executor Status

Build-In Node

1 Idle

2 Idle

Manage nodes and clouds

Refresh status

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	agent1	Linux (amd64)	In sync	4.18 GB	0 B	4.18 GB	64ms
	Built-In Node	Linux (amd64)	In sync	3.83 GB	0 B	3.83 GB	0ms
Data obtained			47 min	47 min	47 min	47 min	47 min

4. After that you install java on Jenkins Agent and the java version is same as Jenkins master java version.

Also, for this project needs docker, so install docker and docker-compose on Jenkins agent.

Give the docker permission to current user.

```
ubuntu@ip-172-31-81-198:~$ sudo usermod -a -G docker $USER
ubuntu@ip-172-31-81-198:~$ sudo reboot
```

5. check the Jenkins agent log, your agent is successfully connected to master or not?

```
Dashboard > Manage Jenkins > Nodes > agent1

[02/04/23 10:29:31] [SSH] Checking java version of Java
[02/04/23 10:29:31] [SSH] java -version returned 11.0.17.
[02/04/23 10:29:31] [SSH] Starting sftp client.
[02/04/23 10:29:31] [SSH] Copying latest remoting.jar...
Source agent hash is 9AB8BCFEE360F2DC648462B7E841A. Installed agent hash is 9AB8BCFEE360F2DC648462B7E841A
Verified agent jar. No update is necessary.
Expanded the channel window size to 4096
[02/04/23 10:29:32] [SSH] Starting agent process: cd "/home/ubuntu" && java -jar remoting.jar -workDir /home/ubuntu -jar-cache /home/ubuntu/remoting/jarCache
Feb 04, 2023 10:29:32 AM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/ubuntu/remoting as a remoting work directory
Feb 04, 2023 10:29:32 AM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to /home/ubuntu/remoting
====[JENKINS REMOTING CAPACITY]====channel started
Remoting version: 3077.vd99cf116da_6f
Launcher: SSHLauncher
Communication Protocol: Standard in/out
This is a Unix agent
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by jenkins.slaves.StandardOutputSupport$ChannelSupport to constructor java.io.FileDescriptor(int)
WARNING: Please consider reporting this to the maintainers of jenkins.slaves.StandardOutputSupport$ChannelSupport
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
Evacuated stdout
Agent successfully connected and online
```

3] After Agent connecting to Master, add the label to Declarative pipeline.

This is the syntax to add label in Declarative pipeline. You can add label in Jenkinsfile.

```
main - react_django_demo_app / Jenkinsfile

CodeWithRushi Update Jenkinsfile
Latest commit 47e4e42 1 hour ago History
A1 contributor

39 lines (35 sloc) 1.06 KB
Raw Blame Edit View

1 pipeline {
2   agent {
3     label "agent1"
4   }
5   stages {
6     stage('code') {
7       steps {
8         git url: "https://github.com/CodeWithRushi/react_django_demo_app.git", branch: "main"
9       }
10    }
11    stage('build') {
12      steps {
13        sh "docker build -t rushi758/django:latest"
14      }
15    }
16    stage('push') {
17      steps {
18        withCredentials([usernamePassword(credentialsId: "dockermhub", usernameVariable: "USERNAME", passwordVariable: "PASSWORD")]) {
19          sh "docker login -u ${{env.USERNAME}} -p ${{env.PASSWORD}}"
20          sh "docker push rushi758/django:latest"
21        }
22      }
23    }
24  }
25 }
```

And do the commit. Using webhooks it's automatically starting the build on Jenkins UI.

CodeWithRushi / react_django_demo_app Public

forked from LondheShubham153/react_django_demo_app

<> Code Pull requests Actions Projects Wiki Security Insights Settings

General Webhooks Add webhook

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

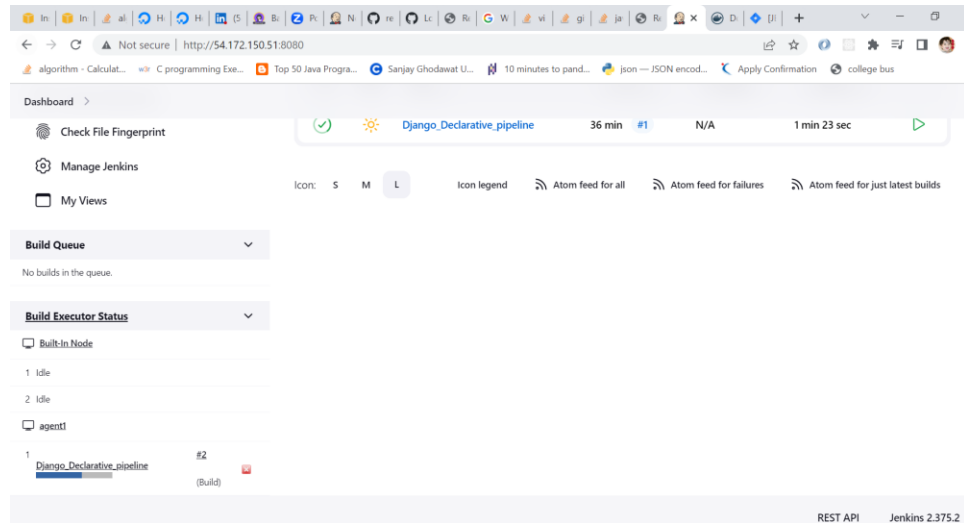
Actions

Webhooks

Environments

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#).

✓ http://54.172.150.51:8080/github-w- (push) Edit Delete



4] After Successfully Run the pipeline, check the logs and full view of pipeline

Dashboard > Django_Declarative_pipeline >

Status

Pipeline Django_Declarative_pipeline

Django_Declarative_pipeline

Stage View

	Declarative: Checkout SCM	Code	Build	push	Test	Deploy
Average stage time: 3s	590ms	49s	13s	81ms	3s	
(Average full run time: ~1min 16s)						
Feb 04 16:02 1 commit	5s	548ms	47s	6s	77ms	2s
Feb 04 19:25 No Changes	706ms	633ms	50s	19s	85ms	3s

Permalinks

- Last build (#2), 1 hr 21 min ago
- Last stable build (#2), 1 hr 21 min ago
- Last successful build (#2), 1 hr 21 min ago

Build History

#2 Feb-5 2023 30:32 AM

#1 Feb-5 2023 9:55 AM

Console Output

Started by Github push by CodeWithHush1

Obtained Jenkinsfile from git https://github.com/CodeWithHush1/react_django_demo_app.git

[Pipeline] Start of Pipeline

[Pipeline] node

Running on agent1 in /home/ubuntu/workspace/Django_Declarative_pipeline

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Declarative: Checkout SCM)

[Pipeline] checkout

Selected Git installation does not exist. Using Default

The recommended git tool is: NONE

No credentials specified

Cloning the remote Git repository

Cloning repository https://github.com/CodeWithHush1/react_django_demo_app.git

> git init /home/ubuntu/workspace/Django_Declarative_pipeline # timeout=10

Fetching upstream changes from https://github.com/CodeWithHush1/react_django_demo_app.git

> git --version # timeout=10

> git --version # 'git version 2.34.1'

> git fetch --tags --force --progress -- https://github.com/CodeWithHush1/react_django_demo_app.git # timeout=10

> git config remote.origin.url https://github.com/CodeWithHush1/react_django_demo_app.git # timeout=10

> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10

Avoid second fetch

Checking out Revision df80a2b16ee48ba9f2fa9a6d2b9d116a3c (refs/remotes/origin/main)

Commit message: "Update Jenkinsfile"

> git rev-parse --verify refs/remotes/origin/main:commit1 # timeout=10

5] Add your Project Port no into your Jenkins agent security Group. Copy the public Ip address and put the port no.

Your application is running

EC2 > Security Groups > sg-067ce41cb2829c935 - launch-wizard-1

sg-067ce41cb2829c935 - launch-wizard-1 Actions

Details

Security group name launch-wizard-1	Security group ID sg-067ce41cb2829c935	Description launch-wizard-1 created 2023-02-04T09:58:59.019Z	VPC ID vpc-09231f74fe21c955
Owner 534497360850	Inbound rules count 2 Permission entries	Outbound rules count 1 Permission entry	

Inbound rules | Outbound rules | Tags

You can now check network connectivity with Reachability Analyzer Run Reachability Analyzer

Inbound rules (2) Manage tags Edit inbound rules

Filter security group rules

<input type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
<input type="checkbox"/>	-	sg-0879b042afca81e3	IPv4	SSH	TCP	22	0.0.0.0/0	-
<input type="checkbox"/>	-	sg-0be0685fa1fc9083	IPv4	Custom TCP	TCP	8001	0.0.0.0/0	-

← → ↻ ⚠ Not secure | http://34.238.246.100:8001

algorithm - Calculat... C programming Ese... Top 50 Java Progra... Sanjay Ghodawat U... 10 minutes to pand... json — JSON encod... Apply Confirmation college bus

Submit