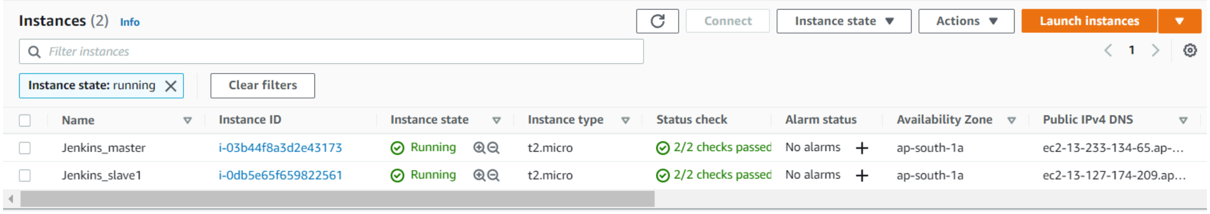
Jenkins installation and master/slave configuration on AWS Cloud

Agenda- In this blog I will show you the easiest and most convenient method of configuring Jenkins's master and slave setup in AWS Cloud.

First let's look at setting up of Jenkins master

Steps:

1. Spin Two Linux instance with minimum CPU and RAM. We will select t2.micro as we don't want to go out of free tier.
   Reference official doc to create an EC2 instance:
   https://docs.aws.amazon.com/efs/latest/ug/gs-step-one-create-ec2-resources.html
2. Name your EC2 Instances like Jenkins_master and Jenkins_slave1 for better understanding.



□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
□	Jenkins_master	i-03b44f8a3d2e43173	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms	ap-south-1a	ec2-13-233-134-65.ap-...
□	Jenkins_slave1	i-0db5e65f659822561	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms	ap-south-1a	ec2-13-127-174-209.ap...

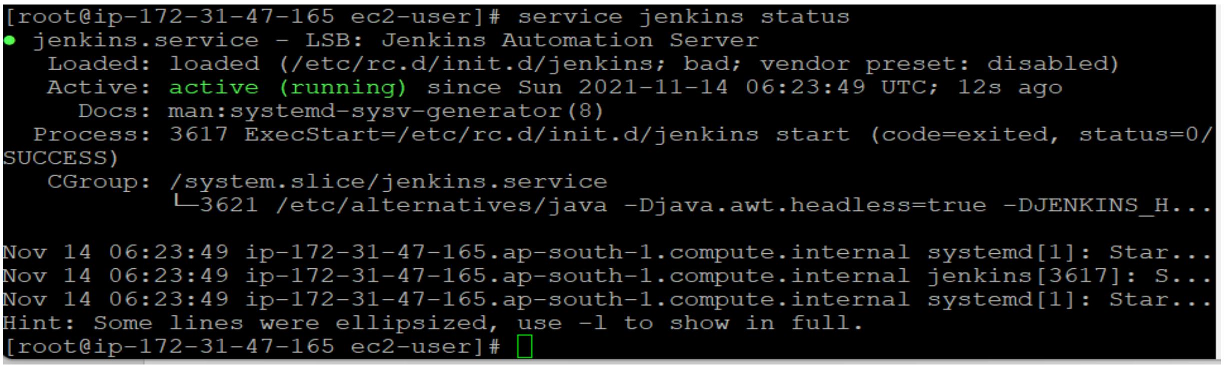
3. Connect to your Jenkins_master instance first using putty.
4. Update your instance with command: yum update -y
5. Now use below commands in sequence for Jenkins installation on Jenkins_master:

   a. sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
   b. sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key

      Note: If you've previously imported the key from Jenkins, the rpm --import will fail because you already have a key. Please ignore that and move on.

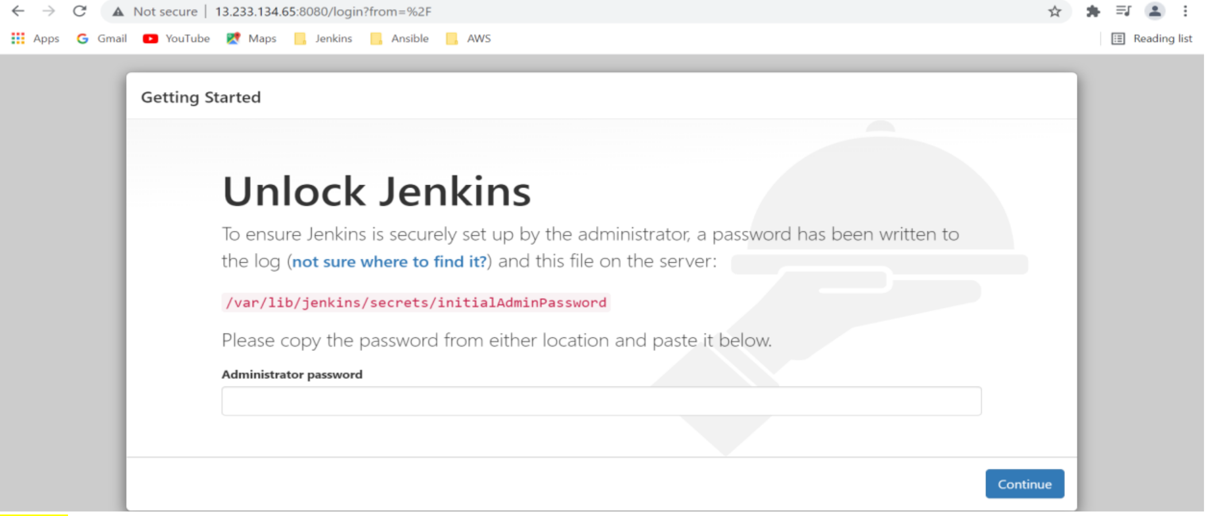
   c. sudo amazon-linux-extras install epel
   d. yum install -y java-1.8.0-openjdk
   e. yum install Jenkins
   f. service jenkins start
   g. service jenkins status

Note: After successful installation of Jenkins, you should be able to see Jenkins service in running status.



```
[root@ip-172-31-47-165 ec2-user]# service jenkins status
● jenkins.service - LSB: Jenkins Automation Server
   Loaded: loaded (/etc/rc.d/init.d/jenkins; bad; vendor preset: disabled)
   Active: active (running) since Sun 2021-11-14 06:23:49 UTC; 12s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 3617 ExecStart=/etc/rc.d/init.d/jenkins start (code=exited, status=0/
SUCCESS)
   CGroup: /system.slice/jenkins.service
           └─3621 /etc/alternatives/java -Djava.awt.headless=true -DJENKINS_H...

Nov 14 06:23:49 ip-172-31-47-165.ap-south-1.compute.internal systemd[1]: Star...
Nov 14 06:23:49 ip-172-31-47-165.ap-south-1.compute.internal jenkins[3617]: S...
Nov 14 06:23:49 ip-172-31-47-165.ap-south-1.compute.internal systemd[1]: Star...
Hint: Some lines were ellipsized, use -l to show in full.
[root@ip-172-31-47-165 ec2-user]#
```

6. Get the public IP address of master and hit that on browser with port 8080 as Jenkins gets deployed by default on port 8080



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

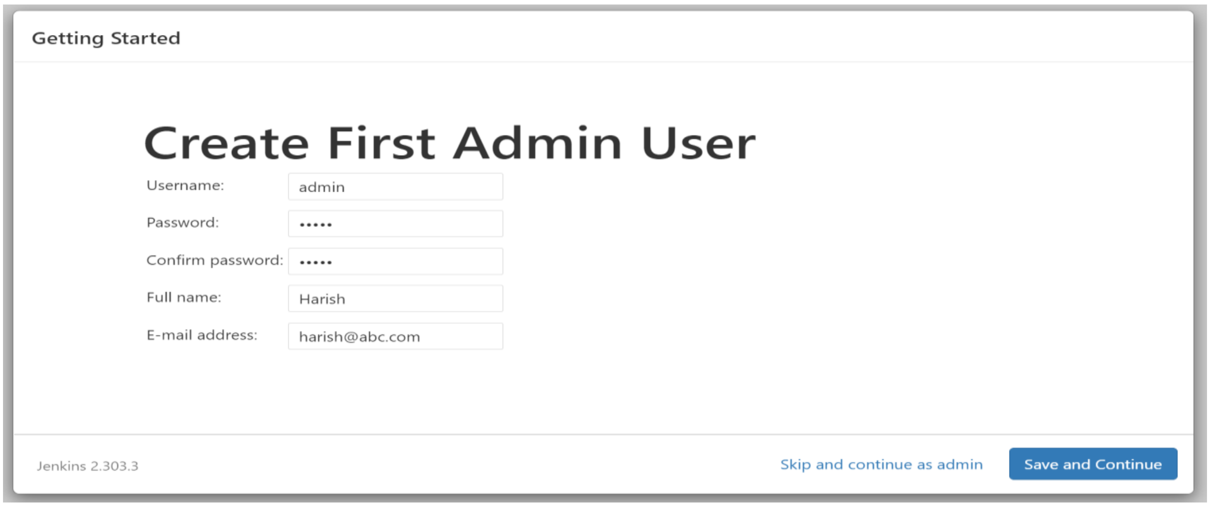
Note: If you want to change the default port then follow below steps else leave it as it is. I am just giving this info in case if its required.

a. cd /etc/sysconfig
b. Edit Jenkins file: vi jenkins
c. here change the port 8080 to your desired port number. For example, to 3333
d. save the file and exit
e. restart Jenkins service

7. Cat the path mentioned in above screenshot and get the password from Jenkins_master instance and paste in Jenkins console.

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

8. You will be prompted with Jenkins customization window, choose appropriate option based on your need. I will go with **install suggested plugins** option.
9. After successful plugin installation you will be prompted with new window and asked for setting up your admin user. Pass appropriate info and create user. After creation of user, you will be able to login onto Jenkins master using the credential.



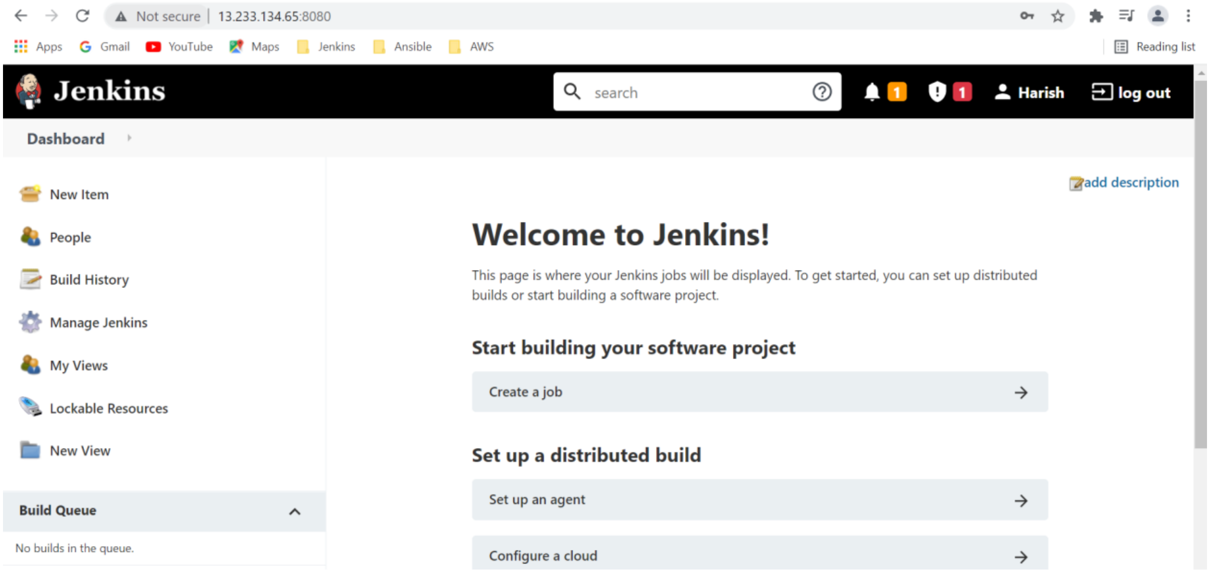
Getting Started

Create First Admin User

Username: admin
Password: •••••
Confirm password: •••••
Full name: Harish
E-mail address: harish@abc.com

Jenkins 2.303.3

10. Now we are done with complete Jenkins installation on master and can login and create jobs on it.



Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Set up a distributed build

No builds in the queue.

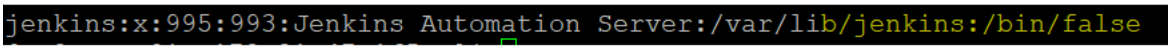
Note: These all steps are mentioned on Jenkins official doc: https://pkg.jenkins.io/redhat-stable/

Steps to achieve master/slave configuration

As we have installed Jenkins on master node, one user must be created with name Jenkins.

This Jenkins user is created in non-interactive mode so we need to change the shell of it.

a. cat /etc/passwd



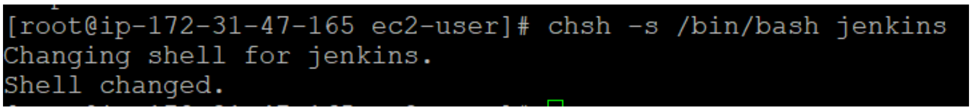
```
jenkins:x:995:993:Jenkins Automation Server:/var/lib/jenkins:/bin/false
```

As you can see /bin/false that means you cannot execute any commands using this user.

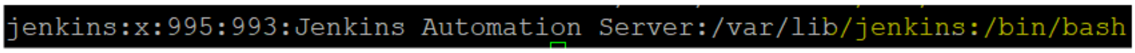
b. So, we need to change the shell of jenkins user using command: chsh -s /bin/bash jenkins

c. If above command doesn't work for you then execute below command and then try to change shell

sudo yum install util-linux-user



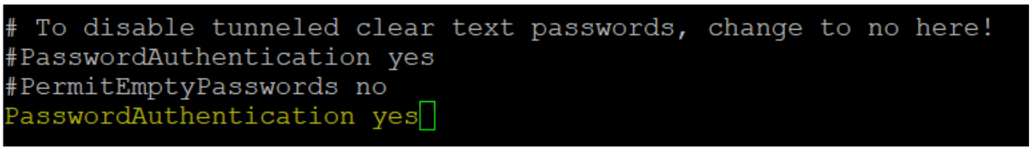
```
[root@ip-172-31-47-165 ec2-user]# chsh -s /bin/bash jenkins
Changing shell for jenkins.
Shell changed.
```



```
jenkins:x:995:993:Jenkins Automation Server:/var/lib/jenkins:/bin/bash
```

Steps to perform on both Jenkins_master and Jenkins_slave1:

1. SSH on Jenkins_master
2. Change password authentication to "yes" as it is set to "no" by default on AWS instance
   - a. cd /etc/ssh/
   - b. vi sshd_config
   - c. change PasswordAuthentication flag from no to yes



```
# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no
PasswordAuthentication yes
```

d. service sshd restart

3. Repeat same steps on Jenkins_slave1 (step 1 and 2)

Now execute below steps on Jenkins_master node

1. Switch to user Jenkins using command: su – jenkins
2. Generate SSH keys using command: ssh-keygen
3. cat id_rsa
4. Copy the complete private key

Now set credentials on Jenkins master console

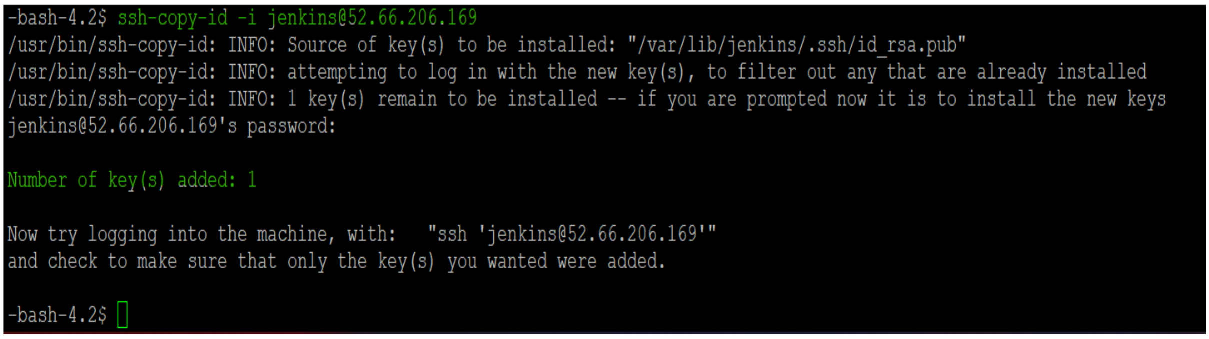
1. Go to manage Jenkins
2. Go to manage credentials
3. Select global domain
4. Add credentials
5. Choose
    a. Kind = SSH Username with private key
    b. Scope = global
    c. Username = jenkins
    d. Private key = Paste your copied private key
6. Click on Ok.

Now go to slave node and perform below steps

1. Install java using command: yum install -y java-1.8.0-openjdk
2. Add user using command: adduser jenkins
3. Set password using command: passwd jenkins

Now go to master node again and execute below one-line command to copy keys on slave node

```
ssh-copy-id -i jenkins@Public_ip_of_slave_node1
for example, ssh-copy-id -i jenkins@52.66.206.169
```



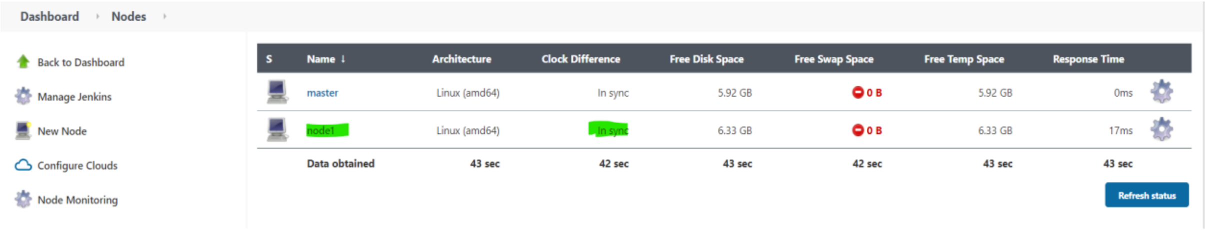
```
-bash-4.2$ ssh-copy-id -i jenkins@52.66.206.169
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/var/lib/jenkins/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
jenkins@52.66.206.169's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'jenkins@52.66.206.169'"
and check to make sure that only the key(s) you wanted were added.

-bash-4.2$ 
```

Note: when asked for password, provide password of Jenkins user which we have created on slave_node1

Now we are left with final step i.e. Add slave node to master node using Jenkins console

Steps:

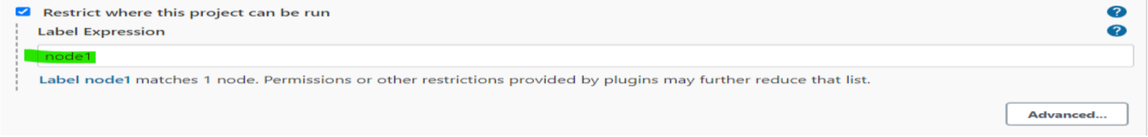
1. Go to manage Jenkins and then manage nodes and clouds
2. Click on New node
3. Give some node name. for example, “Node1”
4. Choose number of executers, we will choose 3 that means 3 parallel jobs can run on our slave node.
5. Remote root directory = /home/Jenkins
6. Labels = (its optional, provide any if you want)
7. Usage = select your usage method. Will select use as much as possible
8. Launch method = Launch agent via ssh
9. Host = IP address of slave node or fully qualified domain name of slave node
10. Credentials = Choose the credentials which we previously added
11. Host Key verification strategy = known host file verification
12. Click on save



S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	master	Linux (amd64)	In sync	5.92 GB	0 B	5.92 GB	0ms
	node1	Linux (amd64)	In sync	6.33 GB	0 B	6.33 GB	17ms
	Data obtained	43 sec	42 sec	43 sec	42 sec	43 sec	43 sec

Let’s test our complete setup with one sample job

1. Create new job
2. While creating job select below option and pass your Jenkins node name

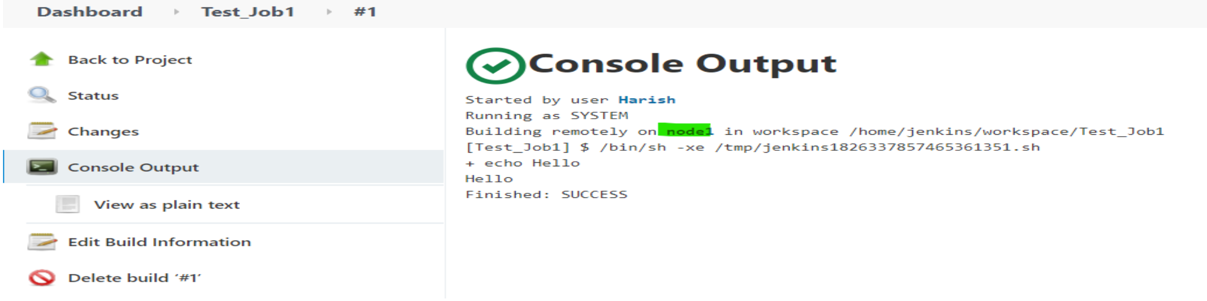


☑ Restrict where this project can be run

Label Expression: node1

Label node1 matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.

3. You can see below console output where we can see job got executed on slave node.



Console Output

```
Started by user Harish
Running as SYSTEM
Building remotely on node1 in workspace /home/jenkins/workspace/Test_Job1
[Test_Job1] $ /bin/sh -xe /tmp/jenkins1826337857465361351.sh
+ echo Hello
Hello
Finished: SUCCESS
```

Thanks for reading my blog
Harish Pawar
DevOps & AWS Cloud Consultant