

Create Master and Slave Machine

Step 1: Create master EC2 machine:

- Install java and jenkins on master node.
- Create jenkins user:
- Create a ssh-keygen on jenkins user
 - Or create ssh-keygen on /var/lib/jenkins/.ssh
- Add necessary permission on ssh keys

```
sudo chmod 700 /var/lib/jenkins/.ssh
sudo chmod 600 /var/lib/jenkins/.ssh/id_rsa
sudo chown -R jenkins:jenkins /var/lib/jenkins/.ssh
```

```
chmod 644 ~/.ssh/id_rsa.pub
```

```
ls -l ~/.ssh/id_rsa.pub [ it must be -rw-r--r-- ]
```

Step 2: Create a slave/worker machine:

- Install java
- Create user like worker.
 - adduser worker
 - passwd worker [Enter any password and remember]

Step 3: Establish connection using SSH

copy the master's jenkins user's public key: **id_rsa.pub** into slave node user[worker] .ssh

```
ssh-copy-id -i ~/.ssh/id_rsa.pub worker@slave_ip
[ pub key copied into slaves user .ssh/authorized.key location ]
```

Connection Test: `ssh -vvv worker@slave-ip`

If you get Permission denied error: Follow below steps

- If password-based login is disabled, enable it:
On **worker node** (172.31.94.120):
 - `sudo nano /etc/ssh/sshd_config`
 - PasswordAuthentication yes
 - PermitRootLogin yes [Its optional]
 - `sudo systemctl restart sshd`

Then try: `ssh-copy-id -i ~/.ssh/id_rsa.pub worker@slave_ip`

Step 4: Connect from master:

ssh worker@slave-ip

Configure Slaves in Jenkins Dashboard

Step 1: Verify Credentials in Jenkins

1. Go to **Jenkins Dashboard → Manage Jenkins → Manage Credentials**.
2. Under "**Global**" credentials, check if the **SSH key credential** (the one used for the slave node) exists.
 - Click **Add Credentials**.
 - **Kind**: SSH Username with Private Key.
 - **Username**: ec2-user (for Amazon Linux) or ubuntu (for Ubuntu). Or user created worker user in slave machine.
 - **Private Key**: Choose **Enter Directly** and **paste the private key** from **Jenkins master** (/var/lib/jenkins/.ssh/id_rsa).
 - **ID**: Set a name (e.g., jenkins-slave-key).
 - Click **Save**.

Step 2:

1. Go to **Jenkins Dashboard → Manage Jenkins → Manage Nodes and Clouds → New Node**.
2. Enter a **name** (e.g., slave1), select **Permanent Agent**, and click OK.
3. Configure:
 - a. **Label**: worker [Same name must be used in jenkinsFile]
 - b. **Remote root directory**: /home/jenkins/
 - i. [This should create on slave node]
 - c. **Usage**: "Use this node as much as possible"
 - d. **Launch method**: "Launch agents via SSH"
 - e. **Host**: <Slave1 Public IP>
 - f. **Credentials**: Select added credential on previous step 1 **or** create new cred
 - i. Click on Jenkins:
 - ii. Add SSH credentials with password:
 - iii. Private Key: ~/var/lib/Jenkins/.ssh/id_rsa or
 1. Login jenkins user: .ssh/id_rsa
 - iv. Username: worker
4. Click **Save & Launch**.

Debug steps:

✓ 1. Verify That the SSH Key Exists in Jenkins

On your Jenkins Master:

The key matches the one on your slave node (`~/.ssh/authorized_keys`)

If missing, generate a new SSH key for Jenkins and add it:

In master

```
sudo -u jenkins  
ssh-keygen -t rsa -b 4096 -f /var/lib/jenkins/.ssh/id_rsa
```

Then, add the public key (`id_rsa.pub`) to the slave node under `~/.ssh/authorized_keys`.

✓ 2. Check Permissions on Jenkins Master

Ensure Jenkins can read the SSH key:

```
sudo chmod 700 /var/lib/jenkins/.ssh  
sudo chmod 600 /var/lib/jenkins/.ssh/id_rsa  
sudo chown -R jenkins:jenkins /var/lib/jenkins/.ssh
```

✓ 3. Verify the Key on the Slave Node:

On the slave, ensure the master public key is in the slaves `authorized_keys` file:

```
cat ~/.ssh/authorized_keys
```

If missing, manually add it:

```
echo "your-public-key-content" >> ~/.ssh/authorized_keys  
chmod 600 ~/.ssh/authorized_keys  
chmod 700 ~/.ssh
```

✓ 4 Ensure Correct SSH Key and Permissions on Master & Slave:

- If the key is missing, **generate a new one**:

```
sudo -u jenkins ssh-keygen -t rsa -b 4096 -f /var/lib/jenkins/.ssh/id_rsa -N ""
```

- Ensure correct permissions:
`sudo chown -R jenkins:jenkins /var/lib/jenkins/.ssh`
`sudo chmod 700 /var/lib/jenkins/.ssh`
`sudo chmod 600 /var/lib/jenkins/.ssh/id_rsa`
`sudo chmod 644 /var/lib/jenkins/.ssh/id_rsa.pub`

✓ 4. Test SSH Manually:

Try SSH from Jenkins Master to Slave:

Be in jenkins user in master node:

`ssh -i /var/lib/jenkins/.ssh/id_rsa worker@54.165.196.151`

✓ 5. Debug with Verbose SSH Logs:

`sudo -u jenkins ssh -vvv -i /var/lib/jenkins/.ssh/id_rsa worker@slave-ip`

Try SSH from Jenkins Master to Slave:

`sudo -u jenkins ssh -i /var/lib/jenkins/.ssh/id_rsa worker@54.165.196.151`

If it asks for a password, key authentication is failing.

If you see "Permission denied (publickey)", the key isn't installed correctly on the slave.

✓ 5. Debug with Verbose SSH Logs

Try connecting from master node:

Su -u jenkins

`ssh -vvv -i /var/lib/jenkins/.ssh/id_rsa worker@slave-ip`

Run this to see detailed errors:

`sudo -u jenkins ssh -vvv -i /var/lib/jenkins/.ssh/id_rsa worker@54.165.196.151`

✓ Solution: Fix SSH Authentication for Jenkins Worker Node

Verify the Worker Node's SSH Access Manually on the Jenkins master node, try connecting manually:

`ssh worker@172.31.94.120`

If it asks for a password, password authentication is required.

If it fails, password-based login might be disabled on the worker node.

If password-based login is disabled, enable it: On worker node (172.31.94.120):

`sudo nano /etc/ssh/sshd_config`

```
PasswordAuthentication yes  
PermitRootLogin yes  
sudo systemctl restart sshd  
ssh worker@172.31.94.120
```