Jenkins Primary/Slave Configuration

This document describes the steps to set up a Jenkins Primary/Slave architecture on AWS EC2 instances, using SSH for communication between the master (primary) Jenkins server and the slave nodes.

# 1. Install Jenkins on Both Primary and Slave Servers

To install Jenkins on both primary and slave servers, follow these steps on each EC2 instance (or other servers):  
  
Step 1: Update the server and install Java:  
sudo apt update  
sudo apt install openjdk-11-jdk  
  
Step 2: Add Jenkins repository key:  
wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -  
  
Step 3: Add Jenkins to the sources list:  
sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'  
  
Step 4: Install Jenkins:  
sudo apt update  
sudo apt install jenkins  
  
Step 5: Start Jenkins:  
sudo systemctl enable jenkins  
sudo systemctl start jenkins

# 2. Configure the Primary Jenkins Master

Step 1: Access Jenkins UI on the primary server (http://<PRIMARY\_SERVER\_IP>:8080). Unlock Jenkins using the initial admin password found at /var/lib/jenkins/secrets/initialAdminPassword.  
  
Step 2: Install recommended plugins and create an admin user.  
  
Step 3: Go to 'Manage Jenkins > Manage Nodes and Clouds > New Node'.  
Step 4: Enter a node name (e.g., 'slave-node') and select 'Permanent Agent'.  
  
Node Configuration:  
 - Set Remote Root Directory (e.g., '/home/jenkins').  
 - Set Number of Executors (define how many jobs this slave can run concurrently).  
 - Choose 'Launch agent via SSH.'

# 3. Configure the Slave Node

Step 1: Ensure the slave server allows SSH connections:  
sudo apt install openssh-server  
  
Step 2: Create a 'jenkins' user on the slave:  
sudo adduser jenkins  
sudo usermod -aG sudo jenkins  
  
Step 3: Generate SSH keys on the master server and copy them to the slave:  
ssh-keygen -t rsa  
ssh-copy-id jenkins@<SLAVE\_SERVER\_IP>

# 4. Set Up Jenkins Agent on the Slave

On the primary Jenkins server UI:  
 - Under 'Manage Nodes', select the new slave node and click 'Launch Agent'.  
 - The agent will connect via SSH.  
  
Alternative method: Manually download and run the agent JAR file on the slave:  
wget <JENKINS\_MASTER\_URL>/jnlpJars/agent.jar  
java -jar agent.jar -jnlpUrl <JENKINS\_URL>/computer/slave-node/slave-agent.jnlp

# 5. Test the Setup

Once the slave node is connected, run sample pipeline jobs on the slave node by specifying the agent label (e.g., 'slave-node').

# 6. Bonus Tips

- Use Docker agents for dynamic scalability.  
- Consider TLS encryption if the environment isn't secure.  
- Implement Node Monitoring Plugins for better resource management.