



## Setting up an On-Prem Jenkins as an Agent to a Cloud Master

In our production environment, our **on-prem Jenkins server was getting heavily loaded** due to multiple concurrent jobs. To improve scalability and stability, I configured it to work as an **agent** connected to our **cloud-hosted Jenkins master** using the **JNLP (Inbound) agent method**.

This setup allowed the **on-prem server to offload builds**, connect **outbound** to the cloud master securely, and maintain better resource utilization without opening inbound firewall ports.

### Problem Statement:

- Our on-prem Jenkins master handled all builds locally. As the number of pipelines and jobs increased, the server experienced:
  - High CPU and memory usage
  - Slower build times and job queue delays
  - Limited scalability
- Instead of moving everything to the cloud, I decided to **keep on-prem compute** but **control it from the cloud Jenkins master**.
- JNLP agents were the perfect fit they connect **outbound to Jenkins**, so no firewall changes were needed.

### Step-by-Step Configuration:

- **STEP1:**  
Verify Jenkins Master is Running On your cloud Jenkins server and ensure Jenkins is accessible.
  - **http://<Jenkins\_public\_IP>:8080** (access the Jenkins server)
  - **systemctl status Jenkins** (Check the status of Jenkins active or not)

## STEP2:

Create a new node on the Cloud Jenkins master

- Navigate to **Manage Jenkins → Nodes → New Node → Permanent Agent**

Set:

- Remote root directory: **/home/Jenkins**
- Launch method: **Launch agent by connecting to the controller**
- Save the node Jenkins will display the node as **offline**.

The screenshot shows the Jenkins 'Nodes' page. At the top, there's a breadcrumb 'Jenkins / Manage Jenkins / Nodes' and a search icon. Below the breadcrumb is a table titled 'Nodes' with columns: S, Name, Architecture, Clock Difference, Free Disk Space, Free Swap Space, Free Temp Space, and Response Time. The table lists three nodes: 'Built-In Node' (Linux (amd64), In sync, 13.97 GiB, 1.01 GiB, 13.97 GiB, 0ms), 'onprem-agent' (Linux (amd64), In sync, 45.04 GiB, 9.31 GiB, 45.04 GiB, 29ms), and 'onprem-server' (Linux (amd64), N/A, N/A, N/A, N/A, N/A). The 'onprem-server' node is highlighted with a red box and has a red 'X' icon next to its name. Below the table, there's a 'Data obtained' row showing '9 min 42 sec' for each column. At the bottom, there's a 'Legend' section with 'Icon: S M L'.

- Click on agent which you have created and Jenkins will display a **jnlpUrl** and **secret** for the agent.

The screenshot shows the Jenkins 'Agent onprem-server' configuration page. The breadcrumb is 'Jenkins / Nodes / onprem-server'. On the left, there's a sidebar with 'Status' (selected), 'Delete Agent', 'Configure', 'Build History', 'Load Statistics', and 'Log'. The main content area is titled 'Agent onprem-server' and has buttons for 'Add description', 'Mark this node temporarily offline', and a help icon. Below the title, there's a section 'Run from agent command line: (Unix)' with a copy icon. The command is: `curl -s0 http://[redacted]/jnlpJars/agent.jar`  
`java -jar agent.jar -url http://[redacted] -name "onprem-server"`  
`-webSocket -workDir "/home/jenkins"`. Below this, there's a section 'Run from agent command line: (Windows)' with a copy icon. The command is: `curl.exe -s0 http://[redacted]/jnlpJars/agent.jar`  
`java -jar agent.jar -url http://[redacted] -secret [redacted] -name "onprem-server"`  
`-webSocket -workDir "/home/jenkins"`. At the bottom, there's a 'Build Executor Status' section showing '0/1' and a dropdown menu with 'onprem-server' (offline).

## STEP3:

Prepare the On-Prem Server

- Install Java and create a dedicated user
  - **sudo apt update**
  - **sudo apt install -y openjdk-21-jdk**

- `sudo adduser Jenkins` (Creates /home/Jenkins)
- `sudo usermod -aG sudo Jenkins` (Give the sudo access)
- `su - Jenkins` (verify Jenkins **id Jenkins**)

#### STEP4:

Download the Agent JAR File

- Copy the URL from above showing agent command line
- `wget http://:8080/jnlpJars/agent.jar` (or) `curl http://:8080/jnlpJars/agent.jar`

Connect the Agent to Cloud Master

- Use the command provided by Jenkins  
`java -jar agent.jar \ -jnlpUrl http://<JENKINS_URL>:8080/computer/onprem-agent/jenkins-agent.jnlp \ -secret <SECRET_KEY> -workDir "/home/jenkins"`

**Note:** If the setup is correct your on-prem node will show ONLINE on the Jenkins dashboard.



S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	13.96 GiB	1.02 GiB	13.96 GiB	0ms
	onprem-agent	Linux (amd64)	In sync	45.04 GiB	9.31 GiB	45.04 GiB	282ms
Data obtained		5 min 25 sec	5 min 25 sec	5 min 25 sec	5 min 25 sec	5 min 25 sec	5 min 25 sec

#### STEP5:

(Optional but Recommended) Run as a Systemd Service. To make sure the agent starts automatically on reboot and stays alive:

- `sudo nano /etc/systemd/system/jenkins-agent.service`

Add:

[Unit] Description=Jenkins JNLP

Agent After=network.target

[Service]

User=jenkins

WorkingDirectory=/home/Jenkins

ExecStart=/usr/bin/java -jar /home/jenkins/agent.jar -jnlpUrl

http://:8080/computer/onprem-agent/jenkins-agent.jnlp -secret -workDir "/home/jenkins"

Restart=always

[Install]

WantedBy=multi-user.target

**Then enable it:**

- sudo systemctl daemon-reload
- sudo systemctl enable jenkins-agent
- sudo systemctl start jenkins-agent

-----\*\*\*-----