



CA - Experiment 8 - Practical Exercise: Set Up a Jenkins CI Pipeline for a Maven Project, Use Ansible to Deploy Artifacts Generated by Jenkins.

This is a **detailed, step-by-step guide** to set up everything from **WSL installation** to **Jenkins Freestyle Project execution** with **Maven, Ansible, SSH setup (without password)**, and **remote deployment**.

🎨 Step-by-Step Guide: Jenkins + Maven + Ansible CI/CD on Windows WSL 🚀

❖ Overview

In this guide, we will: **Install WSL (Ubuntu on Windows)** 🐧

Set up Java, Maven, Git, and Ansible 🔒

Configure SSH connection between WSL & Remote Server 🔑

Set up Jenkins (running on Windows) and integrate it with SSH & Ansible 🛡️

Execute a Jenkins Freestyle Project to deploy JAR files 🎯

● Step 1: Install WSL (Windows Subsystem for Linux)

📌 1.1 Enable WSL

COMMIT TO ACHIEVE

1 Open PowerShell as Administrator and run:

```
1 wsl --install
2
```

2 Reboot your system if required.

📌 1.2 Install Ubuntu on WSL

1 Open Microsoft Store and install **Ubuntu 22.04 LTS**.

2 Launch **Ubuntu** and create a **username** and **password**.

3 **Update packages** inside Ubuntu:

```
1 sudo apt update && sudo apt upgrade -y
2
```

🟡 Step 2: Install Required Tools

📌 2.1 Install Java (JDK 17)

```
1 sudo apt install openjdk-17-jdk -y
2 java -version
3
```

Output should show openjdk version "17.0.x"

📌 2.2 Install Maven

```
1 sudo apt install maven -y
2 mvn -version
3
```

Ensure Maven is installed.

📌 2.3 Install Git

```
1 sudo apt install git -y
2 git --version
3
```



📌 2.4 Install Ansible

```
1 sudo apt update
2 sudo apt install ansible -y
3 ansible --version
4
```

CODERS ARCADE

● Step 3: Set Up SSH Connection (No Password Required)

📌 3.1 Generate SSH Key

Run this inside **WSL (Ubuntu)**:

```
1 ssh-keygen -t rsa -b 4096
2
```

COMMIT TO ACHIEVE

Press Enter for default options.

💡 Why do we write **4096** ? **ssh-keygen -t rsa -b 4096**

The **4096** in the command:

```
1 ssh-keygen -t rsa -b 4096
2
```

specifies the **key length** in bits.

♦ Explanation of the Command:

- **ssh-keygen** → The command to generate an SSH key pair.

- `-t rsa` → Specifies the **type** of key to generate (RSA).
- `-b 4096` → Specifies the **bit length** (i.e., key size).

♦ Why Use 4096 Bits?

Stronger Security:

- A **4096-bit key** is significantly more secure than the default **2048-bit key**.
- It provides better resistance against brute-force attacks.

Long-Term Security:

- As computing power increases, **longer keys** ensure protection for a longer time.
- Many security professionals recommend **4096-bit RSA** for high-security environments.

SSH Best Practices:

- While **2048-bit** is considered secure, **4096-bit** provides extra security without much performance loss.
- Modern systems can handle **4096-bit RSA** efficiently.

♦ When Should You Use 2048 Instead of 4096?

- If performance is a concern (e.g., on embedded devices or old systems).
- If you need faster SSH authentication times.
- However, for **most servers and personal use**, **4096-bit is preferred**.

 **Conclusion:**

Using 4096 makes your SSH keys stronger and more future-proof, ensuring better security for your remote connections. 

3.2 Copy Public Key to Remote Server

```
1 ssh-copy-id ronnie@172.27.158.79
2
```

If prompted, enter the **remote user password**.

COMMIT TO ACHIEVE

Finding the IP Address of the Remote Machine!

The IP address 172.27.158.79 is the **private IP address** of your remote machine (where Ansible is running). Let's break down how you obtained it:

♦ How to Find the IP Address of the Remote Machine?

Since you are running **Windows WSL** and SSH-ing into a remote Linux machine (or another system), you likely found this IP address using one of the following methods:

♦ Method 1: Using `ip a` or `ifconfig` on the Remote Linux Machine

- 1 Log in to the remote machine (where Ansible is deployed).
- 2 Run the following command:

```
1 ip a
```

2

OR

```
1 ifconfig
2
```

3 Look for an entry under `eth0` or `ensXXX` (depending on the network interface).

- Example output:

```
1 2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
   default qlen 1000
2     inet 172.27.158.79/24 brd 172.27.158.255 scope global dynamic eth0
3
```

- Here, `inet 172.27.158.79` is the **IP address** of the machine.

◆ **Method 2: Using** `hostname -I`

Run the following command on the remote machine:

```
1 hostname -I
2
```

This will return the machine's IP address, such as:

```
1 172.27.158.79
2
```

◆ **Method 3: Checking IP on WSL (for Windows Machines)**

If you want to find your WSL (Windows Subsystem for Linux) instance's IP, run:

```
1 ip route | grep default
2
```

Example output:

```
1 default via 192.168.1.1 dev eth0 proto dhcp src 172.27.158.79 metric 100
2
```

Here, `172.27.158.79` is the **WSL machine's IP**.

◆ **How Was It Used in Jenkins?**

- The IP address was used in **Jenkins SSH deployment**:

```
1 scp "C:/Users/Saurav/.jenkins/workspace/Maven-Ansible-Deploy/target/*.jar"
      ronnie@172.27.158.79:/home/ronnie/deployment/
2 ssh ronnie@172.27.158.79 "ansible-playbook /home/ronnie/ansible-deploy/deploy.yml"
3
```

- `ronnie@172.27.158.79` means:

- `ronnie` → The **username** on the remote machine.

- 172.27.158.79 → The **IP address** of the remote machine.

♦ **Why Not Use `localhost` Instead?**

- `localhost` refers to **your own system**, but your **Ansible machine** is on a separate system.
- If the remote machine is on the same network, you **must use its actual IP address** for SSH and SCP.

♦ **What If the IP Changes?**

If your remote machine **restarts** or reconnects, the IP may change (if it's dynamic).

📌 To get a **static IP**, configure a **static IP address** in the network settings or use a **DNS name**.

Summary:

- The **IP address** 172.27.158.79 was obtained by checking the remote machine's network configuration (`ip a` or `hostname -I`).
- It is used for **SSH connection from Jenkins** to copy files and run the Ansible playbook.
- If it changes, you can find the new IP using the same commands and update your Jenkins job accordingly. 🚀

📌 **3.3 Test SSH Login**

```
1 ssh ronnie@172.27.158.79
2
```

If it logs in **without asking for a password**, SSH is correctly set up.

Step 4: Install and Configure Jenkins

📌 **4.1 Install Jenkins on Windows**

1 Download Jenkins MSI from [Jenkins Official Site](#).

2 Install Jenkins and start it as a **Windows Service**.

3 Open Jenkins Web UI at:

```
1 http://localhost:8080
2
```

📌 **4.2 Unlock Jenkins**

1 Copy the **initial admin password** from:

```
1 C:\Users\Saurav\.jenkins\secrets\initialAdminPassword
2
```

2 Paste it into the Jenkins setup page.

📌 **4.3 Install Required Plugins**

♦ Go to **Manage Jenkins** → **Manage Plugins** and install:

Maven Integration Plugin

SSH Agent Plugin

🟡 Step 5: Create the Jenkins Freestyle Project

📌 5.1 Create a New Job

- 1 Go to Jenkins Dashboard → New Item
- 2 Select Freestyle Project → Name it **Maven-Ansible-Deploy**
- 3 Click **OK**

🟢 Step 6: Configure Jenkins Build Steps

📌 6.1 Set Up Source Code Management (Git)

- 1 Under **Source Code Management**, select Git
- 2 Enter your repository URL:

```
1 https://github.com/your-username/MVN-ANS-JEN-CICD.git  
2
```

📌 6.2 Add Maven Build Step

- 1 Go to Build → Add Build Step → Execute Windows Batch Command
- 2 Paste this command:

```
1 cd %WORKSPACE%  
2 mvn clean package  
3
```

📌 6.3 Add SSH Deployment Step

- 1 Go to Build → Add Build Step → Execute shell script on remote host using SSH
- 2 Set **SSH site as:**

```
1 ronnie@172.27.158.79:22  
2
```

3 Enter the command to transfer and deploy JAR:

```
1 scp "C:/Users/Saurav/.jenkins/workspace/Maven-Ansible-Deploy/target/*.jar"  
ronnie@172.27.158.79:/home/ronnie/deployment/  
2 ssh ronnie@172.27.158.79 "ansible-playbook /home/ronnie/ansible-deploy/deploy.yml"  
3
```

Save the job.

ⓘ Steps to add **SSH Credentials** in **Jenkins**.

♦ How to Add SSH in Jenkins for Remote Deployment

To enable **Jenkins** to connect to a remote machine via **SSH**, follow these step-by-step instructions:

Heart Step 1: Install SSH Plugin in Jenkins

- 1 Open **Jenkins** (<http://localhost:8080>).
- 2 Go to **Manage Jenkins** → **Manage Plugins**.
- 3 Click on the **Available** tab and search for "**SSH Pipeline Steps**" or "**Publish Over SSH**".
- 4 Select **Publish Over SSH** and click **Install Without Restart**.
- 5 Once installed, go back to the **Dashboard**.

Heart Step 2: Add SSH Key to Remote Server (Already done for remote machine

Since your Jenkins job runs SSH commands on the remote machine (`ronnie@172.27.158.79`), you need to add the **public key** to the remote server.

- 1 Copy the public key to the remote machine using:

```
1 ssh-copy-id ronnie@172.27.158.79
2
```

OR (if `ssh-copy-id` is not installed)

```
1 cat ~/.ssh/id_rsa.pub | ssh ronnie@172.27.158.79 "mkdir -p ~/.ssh && cat >> ~/ssh/authorized_keys"
2
```

- 2 Verify SSH login without a password:

```
1 ssh ronnie@172.27.158.79
2
```

If successful, Jenkins can now SSH into the remote server without needing a password.

Heart Step 4: Configure SSH in Jenkins (Optional

- 1 Go to **Manage Jenkins** → **Manage Credentials**.
- 2 Under **Stores scoped to Jenkins**, click **(global)**.
- 3 Click **Add Credentials**.
- 4 Choose **Kind: SSH Username with Private Key**.
- 5 Set the following:
 - **Username:** `ronnie`
 - **Private Key:** Choose **Enter directly**, then paste the contents of:

```
1 cat ~/.ssh/id_rsa
2
```

- **Passphrase:** Leave it empty. **6** Click **OK**.

Heart Step 5: Add SSH Host in Jenkins

- 1 Go to **Manage Jenkins** → **Configure System**.
- 2 Scroll down to **Publish Over SSH**.
- 3 Click **Add** under **SSH Servers**.
- 4 Fill in the details:

- **Name:** RemoteServer
- **Hostname:** 172.27.158.79
- **Username:** ronnie
- **Remote Directory:** /home/ronnie/deployment
- **Use Password Authentication, or use a different key:** Unchecked **5** Click **Test Configuration** to verify the connection.

📌 Step 6: Configure Jenkins Job to Use SSH

1 Open your **Jenkins Freestyle Job** (Maven-Ansible-Deploy).

2 Click **Configure**.

3 Under **Build Steps**, add:

- **Execute shell script on remote host using SSH.**

4 Enter the SSH command to copy and deploy the JAR file:

```
1 scp "C:/Users/Saurav/.jenkins/workspace/Maven-Ansible-Deploy/target/*.jar"
ronnie@172.27.158.79:/home/ronnie/deployment/
2 ssh ronnie@172.27.158.79 "ansible-playbook /home/ronnie/ansible-deploy/deploy.yml"
3
```

5 Click **Save**.

📌 Step 7: Test the Setup

1 Run the **Jenkins job**.

2 Check the **Console Output** for SSH connection success.

3 If everything works, Jenkins should:

- Copy the **.jar** file via SCP.
- Trigger the Ansible playbook to deploy the application.

✓ Done! 🎉 Your Jenkins pipeline now connects to the remote server over SSH for deployment. 🚀

COMMIT TO ACHIEVE

● Step 7: Create Ansible Playbook for Deployment

📌 7.1 Create Deployment Directory

On the **remote server**, run:

```
1 mkdir -p /home/ronnie/deployment
2
```

F The **-p** option in the **mkdir** command stands for "**parents**", and it ensures that:

- ✓ If the parent directories in the given path **do not exist**, they will be created automatically.
- ✓ If the directory **already exists**, no error will be thrown.

Example:

```
1 mkdir -p /home/ronnie/deployment
2
```

- If `/home/ronnie` already exists, only `deployment` will be created.
- If neither `/home/ronnie` nor `deployment` exists, both will be created.
- If `deployment` already exists, the command does **nothing** (no error).

This is useful to **avoid errors** when creating directories in scripts or automation! 🚀

📌 7.2 Write Ansible Playbook

1 Create the playbook:

```
1 nano /home/ronnie/ansible-deploy/deploy.yml
2
```

2 Paste the following YAML script:

```
1 ---
2 - hosts: localhost
3   tasks:
4     - name: Ensure deployment directory exists
5       file:
6         path: /home/ronnie/deployment
7         state: directory
8         mode: '0755'
9
10    - name: Copy the JAR file to the deployment folder
11      copy:
12        src: /home/ronnie/deployment/*.jar
13        dest: /home/ronnie/deployment/
14
```

3 Save and Exit (**CTRL + X**, then **Y**, then **Enter**).

📌 7.3 Test Ansible Playbook

```
1 ansible-playbook /home/ronnie/ansible-deploy/deploy.yml
2
```

● Step 8: Run the Jenkins Build 🚀

1 Go to Jenkins Dashboard → Maven-Ansible-Deploy

2 Click Build Now ➔

3 Jenkins will: ✓ Clone your GitHub repo

✓ Build the **Maven project**

✓ Deploy the **JAR file using SSH & Ansible**

4 Check remote deployment:

```
1 ls -l /home/ronnie/deployment
2
```

✓ You should see the **JAR file** inside `/home/ronnie/deployment/` !

Congratulations! You've Successfully Set Up CI/CD! 🏆

- ✓ WSL, Maven, Git, and Ansible installed
- ✓ SSH connection configured (passwordless login)
- ✓ Jenkins job triggers SSH deployment using Ansible
- ✓ Fully automated CI/CD pipeline is up & running 🚀

Additional Resources 📖

- ◆ Jenkins Documentation: [Jenkins User Documentation](#)
- ◆ Maven Documentation: [Official Maven Documentation](#)
- ◆ Ansible Documentation: [Ansible Official Documentation](#)

🚀 All the best! Happy DevOps-ing! 🎉🔥



CODERS ARCADE

COMMIT TO ACHIEVE