## Phase 1: AWS EC2 Startup

1. **Launch 2 Instances:** Dono Ubuntu 24.04 lijiye aur Instance type **t3.medium** select kijiye.
2. **Names:** Ek ka naam `new-r-main` aur dusre ka `new-r-agent` rakhiye.
3. **Security Group:** * `new-r-main`: Port **8080** (Jenkins) aur **22** (SSH) allow kijiye.
    - `new-r-agent`: Port **22** (SSH) allow kijiye (Source mein Main ka Private IP daalna safe hai).

## Phase 2: `new-r-main` Setup (Jenkins Controller)

SSH se Main instance mein login karein aur ye run karein:

1. **Install Java 21:**

```
sudo apt update && sudo apt install -y fontconfig openjdk-21-jre
```

2. Install Jenkins:

```
sudo wget -O /etc/apt/keyrings/jenkins-keyring.asc
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
echo "deb [signed-by=/etc/apt/keyrings/jenkins-keyring.asc] https://pkg.jenkins.io/debian-stable
binary/" | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt update && sudo apt install -y jenkins
sudo systemctl start jenkins && sudo systemctl enable jenkins
```

**Unlock Jenkins:** Browser mein `http://<Main-Public-IP>:8080` par jao aur admin password se setup finish karo.

## Phase 3: `new-r-agent` Setup (The Worker)

SSH se Agent instance mein login karein aur ye commands chalayein:

1. **Install Java 21 & Build Tools:**

```
sudo apt update && sudo apt install -y openjdk-21-jdk git
```

Manual Maven Installation (3.9.12):

```
cd /opt
sudo wget https://dlcdn.apache.org/maven/maven-3/3.9.12/binaries/apache-maven-3.9.12-bin.tar.gz
sudo tar -xvzf apache-maven-3.9.12-bin.tar.gz
sudo mv apache-maven-3.9.12 maven
sudo chown -R ubuntu:ubuntu /opt/maven
# System-wide link (Jo humne pichli baar fix kiya tha)
sudo ln -s /opt/maven/bin/mvn /usr/bin/mvn
```

Workspace Directory:

```
mkdir -p /home/ubuntu/jenkins
sudo chown -R ubuntu:ubuntu /home/ubuntu/jenkins
```

## Phase 4: Connecting the Agent (Jenkins UI)

Ye steps **Main** ke browser dashboard par honge:

1. **Credentials:** `Manage Jenkins > Credentials` mein jaakar `SSH Username with private key` select karein. Username `ubuntu` rakhein aur apni `.pem` key paste karein.
2. **Add Node:** `Manage Jenkins > Nodes > New Node`.
   - **Name:** `new-r-agent`
   - **Remote root directory:** `/home/ubuntu/jenkins`
   - **Label:** `maven-build`
   - **Launch method:** Launch agents via SSH.
   - **Host:** Agent ka **Private IP**.
   - **Host Key Verification:** Non-verifying Verification Strategy.

## Phase 6: Setting the ip permanency without elastic ip

```
sudo mkdir -p /var/lib/jenkins/init.groovy.d
```

```
sudo nano /var/lib/jenkins/init.groovy.d/update-jenkins-url.groovy
```

```
import jenkins.model.*

import java.net.URL

// 1. Get a Token for AWS Metadata (IMDSv2)

def tokenUrl = new URL("http://169.254.169.254/latest/api/token")

def connection = tokenUrl.openConnection()

connection.setRequestMethod("PUT")

connection.setRequestProperty("X-aws-ec2-metadata-token-ttl-seconds", "21600")

def token = connection.getInputStream().getText()

// 2. Use the Token to get the Public IP

def ipUrl = new URL("http://169.254.169.254/latest/meta-data/public-ipv4")

def ipConn = ipUrl.openConnection()

ipConn.setRequestProperty("X-aws-ec2-metadata-token", token)

def publicIp = ipConn.getInputStream().getText()

// 3. Set the Jenkins URL (assuming port 8080)

def newUrl = "http://${publicIp}:8080/"

def jlc = JenkinsLocationConfiguration.get()

jlc.setUrl(newUrl)

jlc.save()

println "--- AUTOMATION: Jenkins URL updated to ${newUrl} ---"
```

```
sudo chown -R jenkins:jenkins /var/lib/jenkins/init.groovy.d/

sudo systemctl restart jenkins
```

`sudo journalctl -u jenkins | grep "AUTOMATION"` (You should see a line that says: `--- AUTOMATION: Jenkins URL updated to http://<YOUR_IP>:8080/ ---`.)

## Phase 6: Testing the Pipeline

Ab ek naya Pipeline job banayein aur ye script use karein jo humne pichli baar "SUCCESS" kiya tha:

```
pipeline {
    agent { label 'maven-build' }
    stages {
        stage('Checkout & Build') {
            steps {
                git 'https://github.com/jenkins-docs/simple-java-maven-app'
                // Final bypass command
                sh 'mvn clean package -DskipTests -Denforcer.skip=true -Dmaven.compiler.source=21 -Dmaven.compiler.target=21'
            }
        }
        stage('Archive') {
            steps {
                archiveArtifacts artifacts: 'target/*.jar', fingerprint: true
            }
        }
    }
}
```