Jenkins Set-up for Ubuntu

Create an AWS Instance

- Choose **Ubuntu 22**
- Network settings: **Allow HTTP, HTTPS**
- Edit network settings: **Add security group rule → All traffic, Anywhere**
- Launch instance
- Run:
- sudo -i

Install Jenkins

1. Open browser → search **Jenkins download (www.jenkins.io)**
2. Scroll down to **Stable (LTS) section** → Click on **Ubuntu**
3. Copy Java installation commands → Paste in AWS CLI
4. Scroll up and copy **Ubuntu Long Term Support (LTS) release commands** → Paste in AWS CLI
5. Check Jenkins status:
6. systemctl status jenkins
7. Get public IP of instance → Open browser:
8. http://<instance-public-ip>:8080
9. Copy path from **Unlock Jenkins** page → Run in terminal:
10. cat <paste-path>
11. Copy password → Paste in browser → Connect
12. Install plugins → Create first admin user → Save and finish

---

Sample Project

Create a New Job

1. **Jenkins Dashboard → New Item**
2. Enter name → Select **Freestyle Project**
3. Configure sections like **General, SCM, Build Triggers, Build Environment, etc.**
4. In **Build Steps**:
    - Select **Execute Shell**
    - Enter:
    - echo "Hello, Hi"

5. Click **Save** → **Apply** → **Build Now** → Check **Console Output**

Continuous Deployment

Configure Build Triggers

- **Every minute:**
    - Job → **Configure** → **Build Triggers** → **Build Periodically** → Schedule: *****
- **Only when code is pushed to repo:**
    - Job → **Configure** → **Build Triggers** → **Poll SCM** → Schedule: *****
- **Trigger on GitHub push:**
    - Job → **Configure** → **Build Triggers** → **GitHub hook trigger for GITScm polling** → Apply → Save
    - On GitHub:
        - **Settings** → **Webhooks** → **Add Webhook**
        - **Payload URL**: http://<instance-public-ip>:8080/github-webhook/
        - **Trigger event**: Just the **push event** → Add Webhook

---

Build a Pipeline Using Jenkins, Maven, and Git

Pipeline Stages:

1. **Clone Git** → 2. **Compile** → 3. **Code Review** → 4. **Unit Testing** → 5. **Package** → 6. **Deploy**

Clone Job

GitHub Repo: https://github.com/Become-DevOps/DevOpsClassCodes.git

Compile Job

1. **Jenkins Dashboard** → **Manage Jenkins** → **Global Tool Configuration**
2. Scroll down to **Maven Installation** → Add Maven:
    - **Name**: my-maven
    - **Install Automatically**: Version 3.9.9
    - Save
3. **New Item** → **Freestyle Job** → **SCM** → **Git** → Paste repo URL
4. **Build Step** → **Invoke top-level Maven targets**:
    - **Maven Version**: my-maven
    - **Goals**: compile

- Save → Build Now

Code Review Job

1. Create a **Freestyle Job**
2. Add **SCM** → **Git** → Paste repo URL
3. **Build Step** → **Invoke top-level Maven targets**:
   - **Goals**: pmd:pmd
4. Install **Warnings Plugin**:
   - **Manage Jenkins** → **Available Plugins** → Search **Warnings** → Install
5. **Configure Code Review Job**:
   - **Post-Build Actions** → **Record Compiler Warnings and Static Analysis Results**
   - **Tool**: PMD
   - **Report File Pattern**: **/pmd.xml
   - Save → Build → View PMD warnings

Test Job

1. Create a **Freestyle Job**
2. Add **SCM** → **Git** → Paste repo URL
3. **Build Step** → **Invoke top-level Maven targets**:
   - **Goals**: test
4. **Configure Post-Build Actions**:
   - **Publish JUnit Test Result Report**
   - **Test Report XML**: target/surefire-reports/*.xml
   - Save → Refresh → View test results

Package Job

1. Create a **Freestyle Job**
2. Add **SCM** → **Git** → Paste repo URL
3. **Build Step** → **Invoke top-level Maven targets**:
   - **Goals**: package
   - Save → Build

Linking Jobs

Upstream Method

1. **Clone Job** → Configure → **Post-Build Actions** → **Build Other Projects** → Compile
2. **Compile Job** → Configure → **Post-Build Actions** → **Build Other Projects** → Code Review
3. Repeat for other stages

Downstream Method

1. **Compile Job** → Configure → **Build Triggers** → **Build After Other Projects Are Built** → Clone
2. **Code Review Job** → Configure → **Build Triggers** → Compile
3. Repeat for other stages

Deploying to Tomcat Server

Install Tomcat on Ubuntu 22

sudo apt-get install -y tomcat9 tomcat9-admin

sudo systemctl start tomcat9

sudo systemctl enable tomcat9

Configure Tomcat Users

1. Edit **tomcat-users.xml**:
2. sudo nano /etc/tomcat9/tomcat-users.xml
3. Add:
4. <role rolename="manager-gui"/>
5. <role rolename="manager-script"/>
6. <role rolename="manager-status"/>
7. <role rolename="admin-gui"/>
8. <user username="admin" password="password" roles="manager-gui,manager-script,manager-status,admin-gui"/>
9. Restart Tomcat:
10. sudo systemctl restart tomcat9
11. Access Tomcat:
12. http://<instance-public-ip>:8080

Deploy WAR to Tomcat

1. Install **Deploy to Container Plugin**:
    - Manage Jenkins → **Available Plugins** → **Deploy to Container** → Install
2. Configure Deployment:

- **Post-Build Action → Deploy WAR/EAR to a container**
- Deploy ****/*.war**
- **Container**: Tomcat 9x Remote
- **Credentials**: Jenkins → Tomcat URL: http://<private-ip-of-qa>:8080

3. Access deployed app:
4. http://<public-ip-of-qa>:8080/qaenv

This guide covers setting up Jenkins, creating jobs, linking them, and deploying a project using Tomcat. Let me know if you need modifications! 🚀