This guide explains the process of creating an Amazon EC2 instance for hosting Apache Tomcat, modifying its configuration, setting up a username and password, and integrating Tomcat with Jenkins for Java project deployments.

1. Launching an EC2 Instance

Step 1: Log in to AWS Console

1. Navigate to the AWS Management Console.
2. Search for and select **EC2**.

Step 2: Launch an EC2 Instance

1. Click on **Launch Instance**.
2. Choose an Amazon Machine Image (AMI), such as:
    - **Amazon Linux 2 AMI** (recommended for most use cases).
3. Select an instance type, e.g., **t2.micro** (free tier eligible).
4. Configure the instance settings, such as network and storage.
5. Add a key pair for SSH access (or use an existing key pair).
6. Configure security groups:
    - Add an **HTTP** rule (port 80) and **Custom TCP Rule** for Tomcat (port 8080).
7. Launch the instance.

Step 3: Connect to the Instance

1. SSH into the EC2 instance:

```
ssh -i your-key.pem ec2-user@<instance-public-ip>
```

2. Installing Java

Apache Tomcat requires Java to run.

1. Update the package repository:

```
sudo yum update -y
```

2. Install Java (Amazon Linux 2 example):

```
sudo yum install java-1.8.0-openjdk -y
```

3. Verify the Java installation:

```
java -version
```

3. Downloading and Setting Up Apache Tomcat

Step 1: Download Tomcat

1. Navigate to the Apache Tomcat Downloads page.
2. Copy the URL of the desired version (e.g., **Tomcat 9**).
3. Download and extract Tomcat on the EC2 instance:
4. wget <tomcat-download-link>
5. tar -xvf apache-tomcat-9.*.tar.gz

```
mv apache-tomcat-9.* tomcat
```

Step 2: Start Tomcat

1. Navigate to the Tomcat directory:

```
cd tomcat/bin
```

2. Start the Tomcat server:

```
./startup.sh
```

3. Access Tomcat via http://<instance-public-ip>:8080.

4. Configuring Tomcat for Authentication

Step 1: Modify context.xml

1. Open the context.xml file for the **Manager** app:

```
nano webapps/manager/META-INF/context.xml
```

2. Comment out the following lines to disable security restrictions:
3. <!-- <Valve className="org.apache.catalina.valves.RemoteAddrValve"
   allow="127\d+\.\d+\.\d+\.\d+|::1" /> -->
4. Save and exit the file (Ctrl+O, Enter, Ctrl+X).

Step 2: Modify tomcat-users.xml

1. Open the tomcat-users.xml file:

nano conf/tomcat-users.xml

2. Add a user with manager-gui and admin-gui roles:
3. <role rolename="manager-gui"/>
4. <role rolename="admin-gui"/>

<user username="admin" password="password" roles="manager-gui,admin-gui"/>

5. Save and exit the file (Ctrl+O, Enter, Ctrl+X).
6. Restart Tomcat to apply changes:
7. ./shutdown.sh

./startup.sh

5. Integrating Tomcat with Jenkins

Step 1: Install Jenkins on EC2

1. Add the Jenkins repository and install Jenkins:
2. sudo yum install -y wget
3. wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
4. sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key

sudo yum install jenkins -y

5. Start Jenkins:

sudo systemctl start jenkins

6. Access Jenkins at http://<instance-public-ip>:8080.

Step 2: Install the Deploy to Container Plugin

1. Log in to Jenkins.
2. Navigate to **Manage Jenkins** > **Manage Plugins**.
3. Search for and install the **Deploy to Container** plugin.

Step 3: Configure Jenkins for Tomcat Deployment

1. Navigate to **Manage Jenkins** > **Configure System**.

2. Under **Deploy to Container** settings, add a new container:
    - **Container Type**: Tomcat 9.x Remote.
    - **Manager URL**: http://<instance-public-ip>:8080/manager/text.
    - **Username**: admin.
    - **Password**: password.

Step 4: Configure a Jenkins Job for Deployment

1. Create a new Jenkins job:
    - Select **Freestyle Project**.
2. In the **Build** section, add the **Deploy War/Ear to Container** post-build action:
    - Specify the WAR file path.
    - Choose the configured Tomcat container.
3. Save and run the job to deploy the Java project.