# PROJECT-1 (MAVEN WEB APP)

List of steps involved in deploying a Maven Web application on Tomcat server.

|  |  |
| --- | --- |
| Description | Command |
| Install updates to linux instance | sudo yum update -y |
| Install openjdk-11 | sudo yum install java-11-openjdk -y |
| Check the java version | java -version |
| Verify the JAVAHOME link | readlink -f $(which java) |
| Install wget utility | sudo yum install git wget -y |
| Verify the git version | git --version |
| Download the latest version of Apache-Maven | **sudo wget** [**https://dlcdn.apache.org/maven/maven-3/3.9.1/binaries/apache-maven-3.9.1-bin.tar.gz**](https://dlcdn.apache.org/maven/maven-3/3.9.1/binaries/apache-maven-3.9.1-bin.tar.gz) **-P /tmp** |
| Extract the maven tar file to ‘/opt’ | sudo tar xvzf /tmp/**apache-maven-3.9.1-bin.tar.gz** -C /opt |
| Rename the extracted directory to ‘maven’ or we can also create a soft link ‘maven’ in ‘/opt’ directory | sudo mv /opt/**apache-maven-3.9.1** /opt/maven  (or)  sudo ln -s /opt/apache-maven-3.9.1 maven |
| Add the environment variables in ‘/etc/profile.d/maven.sh’ file | **sudo vi /etc/profile.d/maven.sh**  **export M2\_HOME=/opt/maven**  **export PATH=${M2\_HOME}/bin:${PATH}** |
| Change the permissions of ‘maven.sh’ file to executable | sudo chmod +x /etc/profile.d/maven.sh |
| Run the ‘maven.sh’ file with source | source /etc/profile.d/maven.sh |
| Now, maven is available and we can verify maven version. | mvn -version |
| Proceeding to download Apache-tomcat latest version and saving it to ‘/tmp’ directory | sudo wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.74/bin/apache-tomcat-9.0.74.tar.gz -P /tmp |
| Extracting the contents of the tarball to ‘/opt’ directory | sudo tar xvzf /tmp/apache-tomcat-9.0.74.tar.gz -C /opt |
| Rename the extracted directory in ‘/opt’ to ‘tomcat’ | sudo mv /opt/apache-tomcat-9.0.74 /opt/tomcat |
| If we are running other application with port 8080 in the same instance we can consider to change the port for tomcat application by editing the ‘server.xml’ file | sudo nano /opt/tomcat/conf/server.xml  # update port number to 8081 , ctrl+O and ctrl+X |
| Create user and group ‘tomcat’ | sudo useradd tomcat |
| Change ownership of ‘/opt/tomcat’ directory to ‘tomcat’ user and group | sudo chown -R tomcat:tomcat /opt/tomcat |
| Make the binary files of tomcat as executable | sudo sh -c 'chmod +x /opt/tomcat/bin/\*.sh' |
| Verify the JAVA\_HOME environment variable to be added to ‘tomcat.service’ file | readlink -f $(which java)  # copy /usr/lib/jvm/java-11-openjdk-11.0.18.0.10-1.el7\_9.x86\_64 |
| Add the following text to ‘tomcat.service’ file with proper JAVA\_HOME variable | sudo nano /etc/systemd/system/tomcat.service  [Unit]  Description=Apache Tomcat Web Application Container  After=network.target  [Service]  Type=oneshot  RemainAfterExit=yes  User=tomcat  Group=tomcat  Environment="JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-11.0.19.0.7-1.el9\_1.x86\_64/"  Environment="JAVA\_OPTS=-Djava.security.egd=file:///dev/urandom -Djava.awt.headless=true"  Environment="CATALINA\_BASE=/opt/tomcat"  Environment="CATALINA\_HOME=/opt/tomcat"  Environment="CATALINA\_PID=/opt/tomcat/temp/tomcat.pid"  Environment="CATALINA\_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC"  ExecStart=/opt/tomcat/bin/startup.sh  ExecStop=/opt/tomcat/bin/shutdown.sh  [Install]  WantedBy=multi-user.target |
| Reload the system daemon | sudo systemctl daemon-reload |
| Enable tomcat service to run at startup | sudo systemctl enable tomcat |
| Start the tomcat service | sudo systemctl start tomcat |
| Install firewalld if not already installed | sudo yum install firewalld -y |
| Enable firewalld to run at startup (good practice) | sudo systemctl enable firewalld |
| Start firewalld service | sudo systemctl start firewalld |
| Add an exception to port 8081 which is presently being used by tomcat application in this project | sudo firewall-cmd --zone=public --permanent --add-port=8081/tcp |
| Reload the firewall to apply the exception to port 8081 | sudo firewall-cmd --reload |
| Edit the tomcat-users.xml file to add new roles ‘admin-gui’ and ‘manager-gui’ and user credentials for both the roles | sudo nano /opt/tomcat/conf/tomcat-users.xml  Add the following roles in the file:  <role rolename="admin-gui"/>  <role rolename="manager-gui"/>  <user username="admin" password="admin" roles="admin-gui,manager-gui"/> |
| Add your system IP into the allow rules for Manager and Host-manager pages of tomcat application | sudo nano /opt/tomcat/webapps/manager/META-INF/context.xml  Add your system IP to the allow list (or .\* at the end of Value) |
| sudo nano /opt/tomcat/webapps/host-manager/META-INF/context.xml  Add your system IP to the allow list (or .\* at the end of Value) |
| Restart tomcat service to apply the updated changes | sudo systemctl restart tomcat |
| Clone the application code from github repository | git clone https://github.com/javabyraghu/maven-web-app.git |
| Switch into the application directory | cd maven-web-app |
| Build the package using maven | mvn clean package |
| Verify if the ‘.war’ file is created in ‘/target’ sub-directory of the application directory | ls ./target/ |
| Copy the ‘.war’ file to ‘/opt/tomcat/webapps’ directory | sudo cp ./target/01-maven-web-app.war /opt/tomcat/webapps/ |
| Now the application can be accessed by the user at the specified URL | Enter URL like:  http://<ifconfig>:8081/01-maven-web-ap |