# Tomcat installation on EC2 instance

### Pre-requisites

1. EC2 instance with Java v1.8.x

### Install Apache Tomcat

1. Download tomcat packages from <https://tomcat.apache.org/download-80.cgi> onto /opt on EC2 instance
2. # Create tomcat directory
3. cd /opt
4. wget http://mirrors.fibergrid.in/apache/tomcat/tomcat-8/v8.5.35/bin/apache-tomcat-8.5.35.tar.gz

tar -xvzf /opt/apache-tomcat-8.5.35.tar.gz

rename mv apache-tomcat-8.5.50 tomcat

1. give executing permissions to startup.sh and shutdown.sh which are under bin.
2. chmod +x /opt/apache-tomcat-8.5.35/bin/startup.sh

shutdown.sh

1. create link files for tomcat startup.sh and shutdown.sh
2. ln -s /opt/apache-tomcat-8.5.35/bin/startup.sh /usr/local/bin/tomcatup
3. ln -s /opt/apache-tomcat-8.5.35/bin/shutdown.sh /usr/local/bin/tomcatdown

tomcatup

tomcatup

#### Check point :

Access tomcat application from browser on prot 8090

* http://<Public\_IP>:8090

1. now application is accessible on port 8090. but tomcat application doesnt allow to login from browser. changing a default parameter in context.xml does address this issue
2. #search for context.xml

find / -name context.xml

1. above command gives 3 context.xml files. comment () Value ClassName field on files which are under webapp directory. After that restart tomcat services to effect these changes
2. tomcatdown

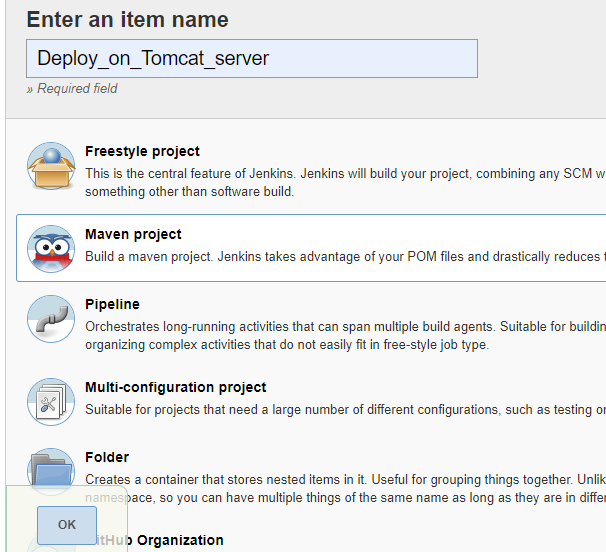
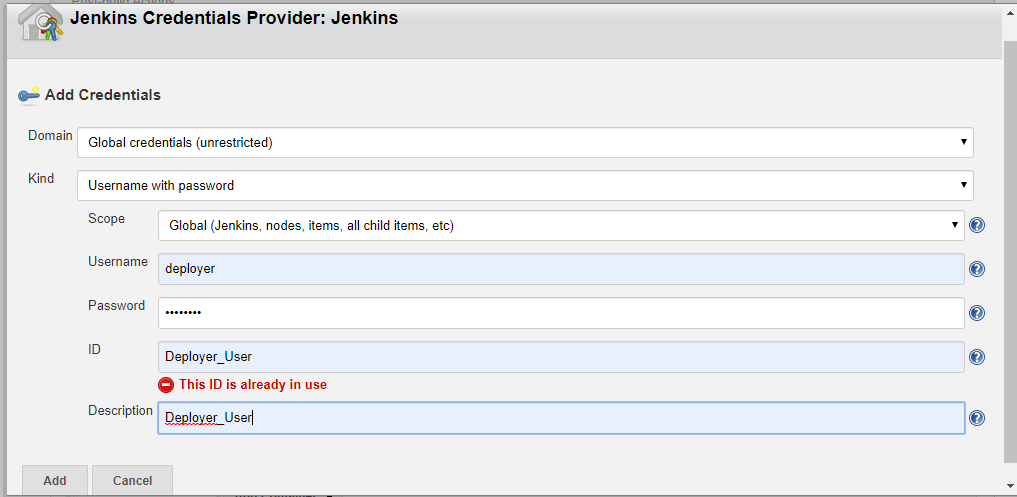
tomcatup

1. Update users information in the tomcat-users.xml file goto tomcat home directory and Add below users to conf/tomcat-user.xml file
2. <role rolename="manager-gui"/>
3. <role rolename="manager-script"/>
4. <role rolename="manager-jmx"/>
5. <role rolename="manager-status"/>
6. <user username="admin" password="admin" roles="manager-gui, manager-script, manager-jmx, manager-status"/>
7. <user username="deployer" password="deployer" roles="manager-script"/>

<user username="tomcat" password="s3cret" roles="manager-gui"/>

1. Restart serivce and try to login to tomcat application from the browser. This time it should be Successful

**DEPLOY A WAR FILE ON TOMCAT Server using Jenkins**

* **We need a plugin called deploy to container.**
  + Manage Jenkins > Jenkins Plugins > available > deploy to container.
  + Create a new job
  + ****
  + Add git repo
  + Under **Post-build Actions** provide the tomcat user details
  + 
  + Apply and run the job.
* **DEPLOY A WAR FILE ON TOMCAT Server using Jenkins Through PollSCM**

Use existing above job and Under POLLSCM need to \* \* \* \* \* and run the job