INSTALLATION OF TOMCAT AND JENKINS ON LINUX AS NON-ROOT USER

I. INSTALL TOMCAT

1. Confirm Java is installed.
   java -version
   It should be version 8 or higher

2. Check which version of tomcat to download that is compatible with installed version of java: https://tomcat.apache.org/whichversion.html

3. Download latest tomcat binary at https://tomcat.apache.org/download-90.cgi

4. Move the tomcat binary to the directory where it is to be installed (e.g. /apps/epsadmin)

5. Uncompress the tomcat binary
   a. tar -xvzf *.tar.gz file

6. Start tomcat: /apps/epsadmin/apache-tomcat-9.0.36/bin/startup.sh

7. Confirm the Apache Tomcat/9.036 home page appears.
   a. Launch browser
   b. Go to <server_name>:8080
   c. The home page file is located at /apps/epsadmin/apache-tomcat-9.0.36/webapps/ROOT/index.jsp

8. Setup Tomcat Web App manager
   a. Edit $CATALINA_HOME/conf/tomcat-users.xml where $CATALINA_HOME=/apps/epsadmin/apache-tomcat-9.0.36/

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>
  <role rolename="manager-gui"/>
  <role rolename="manager-script"/>
  <role rolename="manager-jmx"/>
  <role rolename="manager-status"/>
```

<role rolename="admin-gui"/>
<role rolename="admin-script"/>
<user username="epsadmin" password="mypassword" roles="manager-gui,manager-status,admin-gui,admin-script"/>
<user username="epsadminx" password="my2ndPassword" roles="manager-script,manager-jmx"/>

NOTE: For more details on secure settings, refer to http://<server_name>:8080/docs/manager-howto.html#Configuring_Manager_Application_Access

b. Modify $CATALINA_HOME/webapps/manager/META-INF/context.xml and $CATALINA_HOME/webapps/host-manager/META-INF/context.xml to comment out the entry "<Valve classname …" as shown below:

<!—
<Valve className="org.apache.catalina.valves.RemoteAddrValve" allow="127\.\d+\.d+\.\d+|::1|0:0:0:0:0:0:0:1" />

—>

9. Shutdown and restart tomcat
   a. $CATALINA_HOME/bin/shutdown.sh
   b. $CATALINA_HOME/bin/startup.sh

10. Confirm access to Tomcat Web Application Manager
    a. Go to <server_name>:8080
    b. Click "Manager App" button at the top right
    c. You should be displayed a screen titled "Tomcat Web Application Manager".

II. INSTALL JENKINS

1. Download desired version of Jenkins (e.g. 2.138.1)
   a. wget http://updates.jenkins-ci.org/download/war/2.138.1/jenkins.war
   b. Move the file to /apps/epsadmin

2. Launch Tomcat Web Application Manager, set parameters as follows and then click "Deploy" button in the section labeled "Deploy".

a. In the "Context Path:" field, enter "/jenkins"

b. In the "WAR or Directory path:" field, enter full path to war file (e.g. /apps/epsadmin/jenkins.war)

c. Click "Deploy" and confirm the below response:

Message: OK - Deployed application at the context path [/jenkins]

3. In the "Applications" section, click on the path named "/jenkins".

4. A page should be displayed called "Unlock Jenkins". In this page, you will be presented the full path to a file containing a password. Copy this password and paste it into the "Administrator password" field.

5. If you are displayed a blank page after unlocking Jenkins, then restart Jenkins using the URL http://<server_name>:8080/jenkins. Otherwise, do as instructed and enter the administrator password to unlock Jenkins.

6. Once Jenkins is unlocked, you will be displayed a page to install plugins with one of two options:

a. Option 1: Install suggested plugins

b. Option 2: Select plugins to install

7. Click on "Install suggested plugins". There may be some plugins that fail to install. These failed plugins may require a more recent version of Jenkins.

a. If the failed plugins are not needed, just click "Continue" to proceed.

b. If any of the failed plugins are needed, then you will need to undeploy Jenkins and install a later version of Jenkins. You can do this by going to the Tomcat Web Application Manager and click "Undeploy". Then login to server and remove the /apps/epsadmin/.jenkins directory.

8. When you are satisfied with the Jenkins version and installed plugins (i.e. you have clicked "Continue" to proceed in prior step), you will be presented with a page to create the admin user.

9. Enter the requested information below and click "Save and Continue".

a. Enter Usename:
b. Enter Password:
c. Enter “Confirm Password:”
d. Enter “Full name:”

10. You will be displayed a page titled “Instance Configuration” which shows the Jenkins URL as http://<server_name>:8080/jenkins.

11. Click “Save and Finish”.

12. You will be presented with a page titled “Jenkins is ready!”.

13. Go to the Jenkins URL and start using Jenkins.

III. CONFIGURE JENKINS FOR MYSQL ACCESS

1. As Jenkins administrator, install the MySQL Database Plugin.

2. Login to the MySQL server (note: This is a different server than the Jenkins server).

3. Edit the /apps/mysql/my.cnf file to add the following line in the section [mysqld]:
default_time_zone=‘+00:00’

4. Go to “Manage Jenkins” -> “Configure System”. At the bottom of the page in the section “Global Database”, add a connection by setting the below parameters:
a. Database Type: MySQL
b. Host Name: <fully qualified server name>:<port#>
c. Database: eps_monitor
d. Username: eps_user
e. Password: <password>
f. Validation Query: select count(*) from <tablename>;

5. Click on “Test Connection” button. You should be displayed “OK” indicating successful DB connection. If successful, click “Apply” and then “Save”.

6. Login to the Tomcat/Jenkins server.

7. Locate the mysql-connector-java-8.0.13.jar or equivalent file. It should be in the .jenkins directory.

8. Copy the mysql-connector-java-8.0.13.jar file to the tomcat lib directory.

For example:

cp $HOME/.jenkins/plugins/database-mysql/WEB-INF/lib/mysql-connectory-java-8.0.13.jar $HOME/apache-tomcat-9.0.36/webapps/jenkins/WEB-INF/lib/mysql-connector-java-8.0.13.jar

cp $HOME/.jenkins/plugins/database-mysql/WEB-INF/lib/mysql-connectory-java-8.0.13.jar $HOME/apache-tomcat-9.0.36/lib

9. Test if groovy script can access the MySQL database.

a. Create a new freestyle item named “test_sql”.

b. Select “This project is parametrized”.

c. Click “Add Parameter” button and select “Active Choices” from the drop down options.

d. Enter “ENV” for name.

e. Click “Groovy script” radio button.

f. Enter the Groovy code in the “Groovy Script” field:

```
import groovy.sql.Sql
import java.sql.*
// describe the DB connection params
def db = [
        url:’jdbc:mysql://<server_name>:<db_port#>/eps_monitor’,
        user:’eps_user’,
        password:’mypassword’,
        driver:’com.mysql.jdbc.Driver’  ]
List output = [ ]
try {
        // New connection
        def sql = Sql.newInstance(db.url, db.user, db.password, db.driver)
        // query
        String sqlString = “select distinct env from <tablename>;”
        sql.eachRow(sqlString){ row ->
          output.push(row[0])
        }
        // cleanup
```

```
        sql.close()
} catch (Exception e) {
        output.add(e)
}
if(output_size()==0){
        output.add("Results not found")
}
return output
```

g. Enter into the "Fallback Script" field:

```
return ['error']
```

h. Select "Single Select" for the "Choice Type".

i. Click "Apply" and "Save" buttons.

10. Click "Build with Parameters." You should be displayed a drop down selection menu labeled "ENV".

IV. CONFIGURE JENKINS FOR SQLPLUS ACCESS

1. Login to the Jenkins server and verify sqlplus is installed.

2. As Jenkins admin, add the plugin "Sqlplus Script Runner".

3. Within Jenkins, create Name/Password credentials for the Oracle database.

4. Navigate to your job and select "Configure".

5. Near the bottom of the page, click "Add Build Step" and select "SQLPlus Script Runner" option.

6. Fill in the form as shown below.
    a. Select the applicable credentials.
    b. Instance: <db_server>:1521/<tnsname>

c. Script Type to run: User Defined Script

d. User Defined Script: select sys date from dual;

e. Custom Oracle Home: /apps/oracle/11.2.0.4.0/client

Note: Oracle Home is the Oracle install location on the Jenkins server.

f. Custom SQL *Plus location (sqlplus.exe or sqlplus): /apps/oracle/11.2.0.4.0/client/bin/sqlplus

7. Click “Apply” and “Save”.

8. Click on “Build with Parameters” if you setup a parameterized job.

9. Click on “Build” which launches the job.

10. Monitor the “console output” and verify that you see the SYSDATE displayed.