KeHMIS Kenya Health Management Information Systems

MINISTRY OF HEALTH

Palladium MAKE IT POSSIBLE

DRAFT SOP: OPENMRS UPGRADE FROM v1.X TO VERSION 2.X

KeHMIS Project ©2021

Date: 21st Jul 2021

Objective:

To illustrate the steps to follow to successfully perform installation of KenyaEMR on Ubuntu 20.04. This is to facilitate installation/ upgrade of OpenMRS from version 1.x to 2.x.

Requirements:

i. Ubuntu 20.04 (See the manual)
ii. KenyaEMR modules (OpenMRS 2.x compatible)
iii. OpenMRS webapp (.war file)
iv. Backup of facility KenyaEMR database
v. Java 8 (default)
vi. Tomcat9

PART 1:

Step1:

Upgrade OS to Ubuntu 20.04

Upgrade or Install Ubuntu 20.04LTS (Server or Desktop version). You can obtain the Ubuntu version from here.

Please refer to this link for a guide on how to install Ubuntu Server 20.04LTS

Step 2:

Install Java 8

Run the following commands in the terminal:

```
sudo apt-get update

sudo apt install openjdk-8-jdk

sudo apt-get update
```

Step 3:

Remove Tomcat6 (if exists)

Execute the following commands to uninstall the existing tomcat version from your server. Skip if you are performing fresh installation.

Stop Tomcat6 by running this command in the terminal:

`sudo  service tomcat6 stop` (Stops Tomcat service)

`sudo  apt-get autoremove --purge tomcat6` (uninstalls tomcat6 files)

```
sudo apt-get update
```

Install Tomcat 9

Run the following commands in the terminal:

```
sudo apt install tomcat9
sudo apt install tomcat9-docs tomcat9-examples tomcat9-admin
```

Step 4:

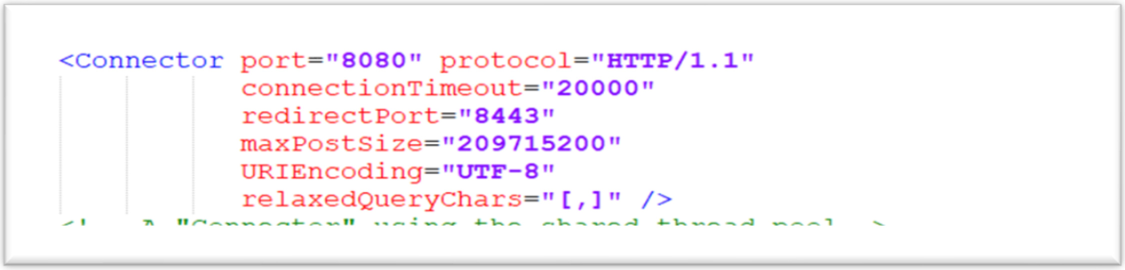
Configure Tomcat 9:

Run the following commands in the terminal:

```
sudo nano /etc/tomcat9/server.xml
```

Under the HTTP Connector, add the following configurations and set the value to 200MB. The final setting should look as below:

```
maxPostSize= “209715200”
URIEncoding="UTF-8"
relaxedQueryChars="[,]"
```



```
<Connector port="8080" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443"
           maxPostSize="209715200"
           URIEncoding="UTF-8"
           relaxedQueryChars="[,]" />
<!-- A "Connector" using the shared thread pool-->
```

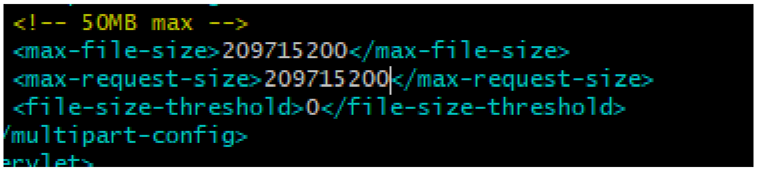
Increase Multipart config in web.xml

Run the following commands in the terminal:

```
sudo nano /usr/share/tomcat9-admin/manager/WEB-INF/web.xml
```

Under the **multipart config**, edit the following config to set it to **200MB**:

```
<max-file-size>209715200</max-file-size>
<max-request-size>209715200</max-request-size>
```



```
<!-- 50MB max -->
<max-file-size>209715200</max-file-size>
<max-request-size>209715200</max-request-size>
<file-size-threshold>0</file-size-threshold>
/multipart-config>
ervlet>
```

On catalina.propterties

Open the catalina.properties file

Tomcat9: *`sudo nano /var/lib/tomcat9/conf/catalina.properties`*

At the tail end of this file, add the following line:

```
org.apache.tomcat.util.buf.UDecoder.ALLOW_ENCODED_SLASH=true
```

Add role config tomcat-users

Run the following commands in the terminal:

```
sudo  nano /etc/tomcat9/tomcat9-users.xml
```

Add the following in the tomcat users.xml file:

```
<role rolename=”tomcat”/>

<role rolename=”manager-gui”/>

<role rolename=”admin-gui”/>

<user username=”admin” password=”PASSWORD” roles=” tomcat, manager-gui,
admin-gui”/>
```

Replace all the quotation marks

Optimize Tomcat

Add the following in `/etc/default/tomcat9` file

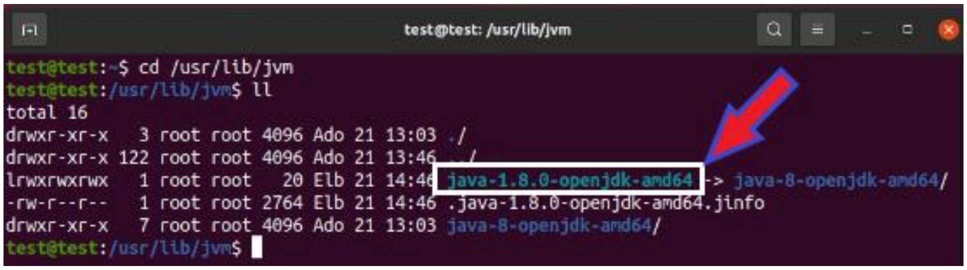
Run command: `sudo nano /etc/default/tomcat9`

Uncomment the line:

`JAVA_HOME=/usr/lib/jvm/java-8-openjdk` (edit to reflect java library name as shown below)

Open a separate terminal and navigate into /`usr/lib/jvm` and list the content to obtain the correct Java library name reference (**e.g.**

- `cd /usr/lib/jvm` [ENTER]
- `ll`



```
test@test:~$ cd /usr/lib/jvm
test@test:/usr/lib/jvm$ ll
total 16
drwxr-xr-x   3 root root 4096 Ado 21 13:03 ./
drwxr-xr-x 122 root root 4096 Ado 21 13:46 ../
lrwxrwxrwx   1 root root   20 Elb 21 14:46 java-1.8.0-openjdk-amd64 -> java-8-openjdk-amd64/
-rw-r--r--   1 root root 2764 Elb 21 14:46 .java-1.8.0-openjdk-amd64.jinfo
drwxr-xr-x   7 root root 4096 Ado 21 13:03 java-8-openjdk-amd64/
test@test:/usr/lib/jvm$ 
```

[ENTER] and obtain the correct java library name – an example shown below:

(NB: Should point to the Java you are using in this case here I am pointing to Java 8)

```
JAVA_OPTS="${JAVA_OPTS} -Xmx2048m -Xms1024m -XX:PermSize=512m -
XX:MaxPermSize=512m -XX:NewSize=256m"
```

NB: The memory allocation above should be based on your RAM size. The above allocations are suitable for a 8GB RAM machine.

Step 5:

Create the OpenMRS folder and set permissions for the tomcat9 user

Run the following commands in the terminal:

```
sudo  mkdir /var/lib/OpenMRS
sudo  chown -R tomcat:tomcat /var/lib/OpenMRS/
sudo  chmod -R 755 /var/lib/OpenMRS*
```

a. Redirect tomcat logs to catalina.out

Edit the file: `/lib/systemd/system/tomcat9.service` and comment out the line: SyslogIdentifier=tomcat9

```
sudo nano /lib/systemd/system/tomcat9.service
```

It should look like this after commenting out: `#SyslogIdentifier=tomcat9`
Add these two lines:

```
StandardOutput=append:/var/log/tomcat9/catalina.out
StandardError=append:/var/log/tomcat9/catalina.out
```

Under **[Service],** add the following line:

```
ReadWritePaths=/var/lib/OpenMRS/
```

Write out (CLtr X) and save the changes

b. Force tomcat9 ReWrite on /var/lib directory

Create **tomcat9.service.d** directory inside etc/systemd/system dir path as follows:

```
sudo mkdir /etc/systemd/system/tomcat9.service.d
```

Inside the created directory, create a blank file names: **logging-allow.conf** as follows

```
sudo nano logging-allow.conf
```

In the file, add the following:

```
ReadWritePaths= /var/lib/OpenMRS
```

(NB: OpenMRS folder must have been already created in var/lib…. Otherwise, it will result into error)
Write out the file and then execute the following commands

```
sudo systemctl daemon-reload
sudo systemctl restart tomcat9
```

To start/stop/restart tomcat9, please type the following commands:

```
     sudo service tomcat9 start
     sudo service tomcat9 stop
     sudo service tomcat9 restart
```

Step 6:
Install MySQL 5.6 on Ubuntu 20.04
Uninstall existing MySQL version if using any other version other than MySQL version 5.6, otherwise skip this step.
Use the following steps to install MySQL 5.6 on Ubuntu 20.04.

```
     sudo add-apt-repository 'deb http://kr.archive.ubuntu.com/ubuntu
     xenial main'
     OR:
     sudo add-apt-repository 'deb http://archive.ubuntu.com/ubuntu trusty
     universe'
```

Then do the following :

```
     sudo apt update
     sudo apt-get install mysql-server-5.6
     sudo touch /var/run/mysqld/mysql.sock
     sudo chown mysql:mysql /var/run/mysqld
     sudo update-rc.d mysql defaults
     sudo service mysql restart
```

Step 7:

Copy the necessary modules and web app file into respective Dirs.

Copy the modules folder and its content into. **OpenMRS** dir:

```
sudo cp -R modules
/var/lib/OpenMRS [ENTER].
```

Copy the .war file into webapp directory:

```
sudo cp openmrs.war /var/lib/tomcat9/webapps
```

Step 8:

- Restart tomcat service and wait until KenyaEMR starts successfully
- Open your browser and enter the correct KenyaEMR URL.
- Wait for all the openmrs schema to be created

NB:

In case of errors while upgrading, go into the upgrade package, navigate into the Script folder, and open the "**database_updates**" file.

Comment out the following lines by adding "#" before the lines:

-
  ```
  DROP index metadatamapping_idx_mdtm_mdclass ON
  metadatamapping_metadata_term_mapping;
  ```

-
  ```
  ALTER TABLE reporting_report_design DROP FOREIGN KEY
  `report_definition_uuid for
  ```

PART 2:

DATABASE RESTORATION AND UPGRADE

After successful platform upgrade, KenyaEMR is likely to load with errors. This could be due to conflicts in concept dictionary versions and the openmrs schema. You will need to restore the facility database and execute the upgrade script to complete.

Step1: Restore database

- Access the facility database and restore it into openmrs schema i.e.
- Log into MySQL : `mysql-uroot -p` [ENTER]
- Drop the existing database: `drop database openmrs;` [ENTER]
- Recreate openmrs database: `create database openmrs;` [ENTER]
- Set openmrs database as default: `use openmrs;` [ENTER]

Restore the facility database in MySQL:

`source /source_database_path/database_name.sql;` [ENTER]

Wait for the database to restore completely.

Step 2: Execute the upgrade script:

- Execute the upgrade script provided with the latest KenyaEMR upgrade package.
- Copy the upgrade package to Home directory (or any other of your choice)
- Open the Terminal (`Ctr+Alt+T`)
- CD into the upgrade package directory.
- Execute the upgrade script: `sh setup_script.sh` [ENTER]
- Wait for the upgrade to compete.

Step 3: Load KenyaEMR Page:

- Open the KenyaEMR page or the browser on http://server_address:8080/openmrs [ENTER]
- Log in to OpenMRS using Admin Account
- Confirm and execute all the database updates.
- Wait until all the updates are completed and refresh the page to load KenyaEMR.

THE END