CiscoTP-OpenIoT Installation on 64 bit Ubuntu 14.04.3 LTS

This is a step by step guide for installing and running the server
Date: 15 December 2015 Version 1.0

Server Info:
DNS: srvgal89.deri.ie
OS: Ubuntu 14.04.3 LTS

MemTotal:	65967448 kB
MemFree:	55088924 kB
MemAvailable:	64171972 kB
Buffers:	309820 kB
Cached:	8588504 kB
Total Storage:	1TB

Step 1:
SSH from you Mac or other machines and give your password

```
$ ssh ciscotp@srvgal89.deri.ie
Password: ********
```

Step 2:
Enter as a sudo user

```
$ sudo su –
$ apt-get update
```

Step 3:
Check the some software version available or not. If not install that software.

Check Java

```
$ java –version
```

You need minimum JDK version 1.7+ and Set JAVA_HOME in .bashrc file.

How to install Java:

```
$ apt-add-repository ppa:webupd8team/java
$ apt-get update
$ apt-get install oracle-java7-installer
```

How to set Java home:

```
$ echo "export JAVA_HOME=/usr/lib/jvm/java-7-oracle" >> ~/.bashrc
$ echo "export PATH=$PATH:JAVA_HOME/bin" >> ~/.bashrc
```

Check Maven

```
$ mvn –v
```

The maven version should be 3.2.5+

How to install Maven3:

```
$ apt-get purge maven maven2 maven3
$ apt-add-repository ppa:andrei-pozolotin/maven3
$ apt-get update
$ apt-get install maven3
```

Check SVN

```
$ svn –version
```

There should be a latest svn installed

How to install SVN:

```
$ apt-get install subversion
```

Check Git

```
$ git version
```

There should be a latest git installed

How to install Git:

```
$ apt-get install git
```

Please double check all the software version again.

Step 4:
Make a project directory and go to this directory

```
$ mkdir /usr/adm
$ cd /usr/adm/
```

Step 5:
Download the CiscoTP OpenIoT project from subversion

```
$ svn co link
```

Step 6:
Install virtuoso stable version and follow the link bellow:
https://github.com/OpenIotOrg/openiot/wiki/InstallingVirtuosoOpensource7Ubuntu

How to install virtusoso:

```
$ git clone https://github.com/openlink/virtuoso-opensource.git Virtuoso-Opensource
$ cd Virtuoso-Opensource git checkout stable/7
$ apt-get install libtool gawk gperf autoconf automake libtool flex bison m4 make openssl libssl-dev
$ ./autogen.sh
$ ./configure
$ make
$ make install
```

Add new user for virtuoso-opensource directory:

```
$ useradd virtuoso --home /usr/local/virtuoso-opensource
$ chown -R virtuoso /usr/local/virtuoso-opensource
```

Check port for virtuoso: (default is 8890)

$ nano /usr/local/virtuoso-opensource/var/lib/virtuoso/db/virtuoso.ini
Change some defaults as:
ServerPort = 8007
ServerName = virtuoso (SRVGAL89)
DefaultHost = localhost:8007
;DefaultGraph = http://localhost:8007/dataspace
;ImmutableGraphs = http://localhost:8007/dataspace

How to run virtuoso:
$ cd /usr/local/virtuoso-opensource/var/lib/virtuoso/db
$ sudo -H -u virtuoso ../../../../bin/virtuoso-t -f &

Go to http://srvgal89.deri.ie:8007/conductor/ and login as user=dba and password=dba

Change user permissions:

i. Go to System Admin -- > User Accounts -- > users
ii. Edit SPARQL user
iii. Add SPARQL_SELECT and SPARQL_UPDATE as account role
iv. Edit dba user
v. Add SPARQL_SELECT, SPARQL_UPDATE, administrators as account role
vi. Give read and write permissions to dba.

Create new Graphs:

i. Go to Linked Data -- > SPARQL
ii. `CREATE GRAPH <http://lsm.deri.ie/OpenIoT/sensormeta#>`
iii. `CREATE GRAPH <http://lsm.deri.ie/OpenIoT/sensordata#>`
iv. `CREATE GRAPH <http://lsm.deri.ie/OpenIoT/eventcalender#>`
v. `CREATE GRAPH <http://lsm.deri.ie/OpenIoT/functionaldata#>`

If you want to use different graph names and different credentials, make sure to also change those in the configuration file jboss/standalone/configuration/openiot.properties

Virtuoso should now be ready for use by LSM.

Step 7:
Install stable Jboss server
$ cd /usr/adm
$ wget http://download.jboss.org/jbossas/7.1/jboss-as-7.1.1.Final/jboss-as-7.1.1.Final.tar.gz
$ tar xfvz jboss-as-7.1.1.Final.tar.gz
$ cd /usr/adm/jboss-as-7.1.1.Final

Start and Stop jboss:
$ screen ./standalone.sh -Djboss.bind.address=0.0.0.0 -Djboss.bind.address.management=0.0.0.0&

```
$ ./standalone.sh -Djboss.bind.address=0.0.0.0 -
Djboss.bind.address.management=0.0.0.0 > log_cisco_10122015.out 2> jboss.err
< /dev/null &
$ tail -f log_cisco_10122015.out
$ netstat -ntlp | grep LISTEN
```

http://srvgal89.deri.ie:8080/

```
$./jboss-cli.sh --connect command=:shutdown
```

Step 8:
Build **utils.commons**

```
$ cd
/usr/adm/ciscoioe_oldOpenIoTwithLocation/utils/utils.commons
$ cp src/main/resources/properties/openiot.properties
/usr/adm/jboss-as-7.1.1.Final/standalone/configuration/
$ nano /usr/adm/jboss-as-
7.1.1.Final/standalone/configuration/openiot.properties
```

Edit the ip and ports and other parameters:

```
scheduler.core.lsm.sparql.endpoint=http://srvgal89.deri.ie:8007/sparql
scheduler.core.lsm.remote.server=http://srvgal89.deri.ie:8080/lsm-
light.server/
sdum.core.lsm.sparql.endpoint=http://srvgal89.deri.ie:8007/sparql
sdum.core.lsm.remote.server=http://srvgal89.deri.ie:8080/lsm-light.server/
lsm-
light.server.connection.url=jdbc:virtuoso://srvgal89.deri.ie:1111/log_enable=2
lsm-light.server.connection.username=dba
lsm-light.server.connection.password=dba
```

```
$ mvn clean package install
```

Start Jboss:

```
$ ./standalone.sh -Djboss.bind.address=0.0.0.0 -
Djboss.bind.address.management=0.0.0.0 > log_cisco_10122015.out 2> jboss.err
< /dev/null &
$ tail -f log_cisco_10122015.out
```

Step 9:
Deploy lsm.light-server

```
$ cd /usr/adm/ciscoioe_oldOpenIoTwithLocation/modules/lsm-light/lsm-light.server
$ nano src/main/java/org/openiot/lsm/websocket/server/Constant.java
```

Change CQELSHOME = "/usr/adm";

```
$ nano src/main/java/org/openiot/lsm/reasoning/data/Constants.java
```

```
aspURI = "/usr/adm/ciscoioe_oldOpenIoTwithLocation/modules/lsm-light/lsm-
light.server/Reasoning/";
queryServerURI = "ws://srvgal89.deri.ie:8002/websockets/query";
virtuosoURI = "http://srvgal89.deri.ie:8007/sparql";
```

```
$ nano src/main/java/org/openiot/lsm/websocket/server/QueryServerEndPoint.java
```

```
QueryExecution qexec = QueryExecutionFactory.sparqlService(
                    "http://srvgal89.deri.ie:8007/sparql", query);
QueryExecution qexec = QueryExecutionFactory.sparqlService("http://srvgal89.deri.ie:8007/sparql", query);
```

Deploy Additional Jars:

```
$ cd /usr/adm/ciscoioe_oldOpenIoTwithLocation/additionalJars/QueryProcessing/
$ ./deployJars.sh
Input: /root
```

```
$ cd /usr/adm/ciscoioe_oldOpenIoTwithLocation/modules/lsm-light/lsm-light.server
$ mvn clean package jboss-as:deploy
```

http://srvgal89.deri.ie:8080/lsm-light.server/

Step 10:

```
$ cd /usr/adm/ciscoioe_oldOpenIoTwithLocation/modules/lsm-light/lsm-light.client
$ mvn clean package install
```

Step 11:

```
$ cd /usr/adm/ciscoioe_oldOpenIoTwithLocation/modules/scheduler/scheduler.core
```

```
$ mvn clean package jboss-as:deploy or mvn jboss-as:deploy
```

http://localhost:8080/scheduler.core/rest/services

Step 12:

```
$ cd modules/sdum/sdum.core directory
```

```
$ mvn clean package jboss-as:deploy or mvn jboss-as:deploy
```

Step 13:

```
$ cd usr/adm/ciscoioe_oldOpenIoTwithLocation/ui/ui.requestCommons
```

```
$ mvn clean package install
```

Step 14:

```
$ cd ui/ui.requestDefinition
```

```
$ mvn clean package jboss-as:deploy or mvn jboss-as:deploy
```

http://localhost:8080/ui.requestDefinition/

Step 15:

```
$ cd usr/adm/ciscoioe_oldOpenIoTwithLocation/ui/requestPresentation
```

```
$ mvn clean package jboss-as:deploy or mvn jboss-as:deploy
```

http://localhost:8080/ui.requestPresentation/

Step 16:

```
$ cd usr/adm/ciscoioe_oldOpenIoTwithLocation/modules/sdum/sdum.client

$ mvn clean package install
```

Step 17:

```
$ cd
usr/adm/ciscoioe_oldOpenIoTwithLocation/modules/scheduler/scheduler.client
$ mvn clean package install
```

Step 18:

```
$ cd /usr/adm/ciscoioe_oldOpenIoTwithLocation/modules/x-gsn/
$ nano /root/.m2/repository/org/codehaus/groovy/groovy-all/1.7.1/groovy-all-1.7.1.pom
```

Make some changes like as bellow:

```
<!--
    <repositories>
        <repository>
            <id>jansi</id>

<url>http://jansi.fusesource.org/repo/release</url>
            <snapshots>
                <enabled>false</enabled>
            </snapshots>
            <releases>
                <enabled>true</enabled>
            </releases>
        </repository>
    </repositories>
-->

<dependency>
            <groupId>org.fusesource.jansi</groupId>
            <artifactId>jansi</artifactId>
            <version>1.11</version>
</dependency>
```

```
$ mvn clean package install
$ nano conf/lsm_config.properties
endPoint=http://srvgal89.deri.ie:8007/sparql
$ nano conf/httplistener_config.properties
httpListenerPort=8006
$ nano conf/gsn.xml
```

Change port to 8001

Start and Stop x-gsn:

```
$ ./gsn-start.sh
$ tail -f gsn_consoleOutput.log
```

http://srvgal89.deri.ie:8001/

```
$ ./gsn-stop.sh
```

Step 19:

```
   i.    Login to OpenMeetings
   ii.   Create new user
   iii.  Login to android app
   iv.   Register all sensors at first
   v.    Create meeting even
   vi.   Enter in the meeting room
   vii.  Get notifications
```