LUX 2.7 Installation Guide

Last Update: 5/8/18

1 Prerequisite Software Downloads

LUX Component	Required Hardware
Engine Server	16 CPU / 32GB RAM / 72GB HDD
User Interface Server	8 CPU / 16GB RAM / 72GB HDD
Database Server	8 CPU / 16GB RAM / 4TB HDD
	(SSD preferred)
Alert Analytic Server (optional)	16 CPU / 32GB RAM / 72GB HDD

Software	Recommended Version	Download URL
CentOS or RHEL Linux	7.x	https://www.centos.org/download/
JDK	1.8.x	http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
MySQL	5.7.22	https://www.mysql.com/downloads
MongoDB	3.4.x-3.6.x	https://www.mongodb.com/download-center#community
HBASE (optional)	5.5+	https://www.cloudera.com/downloads/quickstart_vms/5-13.html
ActiveMQ (optional)	5.3+	http://activemq.apache.org/download.html

2 Prerequisite Software Installation

2.1 JDK 1.8

1. Download either a Oracle JDK or OpenJDK 1.8 .rpm package.
    a. (ex. Oracle: `jdk-8u121-linux-x64.rpm`).
2. `sudo rpm -Uvh jdk-<VERSION>-linux-x64.rpm`
3. If multiple JDKs have been installed, it is possible that alternatives may contain conflicting configurations. Perform the following steps to verify that the everything is set up correctly:
    a. `sudo /usr/sbin/alternatives --config java`

i. Find which number is assigned to Java 1.8

b. Insert the number assigned to the Java 1.8 install to the end of the following command
   i. `sudo /usr/sbin/alternatives --install /usr/bin/java java /usr/java/default/bin/java 4`
      1. Substitute appropriate number for "`4`"

c. Choose the 1.8 alternative

2.2 MYSQL

1. Remove mariadb if installed: `sudo rpm -e --nodeps mariadb-libs`
2. Substitute the following `<VERSION>` tags with the appropriate MySQL version (i.e. 5.7.22)
3. Download a LUX compatible version of the MySQL .rpm bundle:
   - `mysql-5.7.22.x86_64.rpm-bundle.tar`
4. `tar xf mysql-5.7.22.x86_64.rpm-bundle.tar`
5. `sudo rpm -Uvh mysql-community-common-5.7.22.x86_64.rpm` (depending on your system configuration, you may need to use the `--force` or `--nodeps` options for commands 4-7)
6. `sudo rpm -Uvh mysql-community-libs-5.7.22.x86_64.rpm`
7. `sudo rpm -Uvh mysql-community-client-5.7.22.x86_64.rpm`
8. `sudo rpm -Uvh mysql-community-server-5.7.22.x86_64.rpm`
9. `sudo semanage fcontext --add --type mysqld_db_t '/var/lib/mysql'`
   a. (if `semanage` is not available, `sudo yum install policycoreutils-python`)
10. `sudo restorecon -r /var/lib/mysql`
11. `sudo chmod -R 755 /var/lib/mysql`
12. `sudo chown -R mysql.mysql /var/lib/mysql`
13. `sudo mysqld --initialize`
14. `sudo chown -R mysql:mysql /var/lib/mysql`
15. Add the following lines to `/etc/my.cnf`:

```
[client]
default-character-set = utf8mb4
[mysql]
default-character-set = utf8mb4
[mysqld]
character-set-server = utf8mb4
collation-server = utf8mb4_unicode_ci
```

16. `sudo systemctl start mysqld`
17. `sudo systemctl enable mysqld.service`
18. `sudo grep 'temporary password' /var/log/mysqld.log`
19. `mysql_secure_installation`

20. Enter the temporary password from above, and change the password. This password will be used later in section 3.3.1. By default password validation is enabled, so your root password must meet the minimum complexity described below:

Variable Name	Value
validate_password_check_user_name	off
validate_password_dictionary_file	
validate_password_length	8
validate_password_mixed_case_count	1
validate_password_number_count	1
validate_password_policy	MEDIUM
validate_password_special_char_count	1

If you then choose to use a simpler password, you may (after entering a complex password) remove the password policy plugin and change the root password.

19. You will be prompted to answer a number of questions pertaining to the mySQL installation. Answer them according to how your environment will be configured.

2.3 MongoDB or HBASE

MongoDB **or** HBASE are required. Most installations will utilize MongoDB. Follow instructions from section 2.3.1 or 2.3.2 depending on which configuration matches your environment.

2.3.1 MongoDB

Mongo, by default, will store its database files in /var/lib/mongodb. If /var/lib is in a partition that does not have a lot of disk space, you will want to move mongodb to another partition. The instructions below for “From Tarball” assume you’re going to move the mongodb folder to /data/mongodb. The YUM installs do not.

From Yum

1. Substitute the following `<VERSION>` tags with the appropriate MongoDB version (i.e. `3.4.4`)
2. Create `/etc/yum.repos.d/mongodb-org-<VERSION>.repo` with the following content:

```
[mongodb-org-<VERSION>]
name=MongoDB Repository
```

```
baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/<VERSION>/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-<VERSION>.asc
```

3. `sudo yum install -y mongodb-org`
4. `sudo semanage port -a -t mongod_port_t -p tcp 27017`
5. (Optional) Set SELinux to `permissive` mode in `/etc/selinux/config` by setting the `SELINUX` setting to `permissive`
    a. `SELINUX=permissive`
6. `sudo systemctl start mongod`
7. `sudo systemctl enable mongod.service`

From Tarball

1. For CentOS: download `mongodb-linux-x86_64-<VERSION>.tgz`
    a. (ex. `mongodb-linux-x86_64-rhel70-3.4.4.tgz)`
2. `tar -zxvf mongodb-linux-x86_64-<VERSION>.tgz -C /usr/local`
3. `sudo ln -s /usr/local/mongodb-linux-x86_64-<VERSION> /usr/local/mongodb`
4. Create the file `/etc/profile.d/mongodb.sh` and insert `export PATH=$PATH:/usr/local/mongodb/bin` as the first line.
5. `. /etc/profile`
6. Create a data folder: `sudo mkdir -p /data/db`
7. `sudo useradd -c "MongoDB Service" -d /usr/local/mongodb -M -r -U mongo`
8. `sudo chown -R mongo:mongo /usr/local/mongodb/*`
9. If needed for a data volume: `sudo chown -R mongo:mongo /data/*`
10. **If using RHEL/CentOS 7**:
    a. Create the file `/lib/systemd/system/mongod.service` with the following content:

```
[UNIT]
Description=An object/document-oriented database
Documentation=man:mongod(1)
After=network.target

[Service]
User=mongo
Group=mongo
ExecStart=/usr/local/mongodb/bin/mongod

[Install]
WantedBy=multi-user.target
```

    b. `sudo systemctl daemon-reload`
    c. `sudo systemctl enable mongod`
    d. Start MongoDB using `sudo systemctl start mongod`
11. **If using RHEL/CentOS 6/Amazon Linux:** Create the file `/etc/init.d/mongod` with the content from appendix A.1.

a. `sudo touch /etc/init.d/mongod`
b. `sudo chkconfig --add mongod`
c. Start MongoDB using `sudo service mongod start`

Check Mongo configuration via mongod.conf:

1. Ensure mongod is listening on all interfaces.
    a. Edit `/etc/mongod.conf` and find the line that says `Listen to local interface only. Comment out to listen on all interfaces`. Make sure that the next line is either commented out, or specifies `bindIp=0.0.0.0` so that mongod does indeed listen on all interfaces.
    b. Do not put the Application Server IP in `bind_ip` option. This `bind_ip` option tells MongoDB to accept connections from which local network interfaces, not which "remote IP address".
2. If needed: Enable Mongo Security Authorization.
    a. Edit `/etc/mongod.conf` and find the line that says `#security.` Uncomment the `#security` line and add the line "`authorization: enabled`"
3. Verify that any additional configuration settings within /etc/mongod.conf are configured correctly for your particular LUX installation.

2.3.2 HBASE

1. Download the most current HBASE Quick-Start Virtual Machine Image.
2. From the desktop, run the launch cloudera enterprise trial.
3. Login to web console.
4. Verify that the virtual machine time matches your system time
    a. If it does not, then verify that ntpd is configured correctly.
5. Restart the cluster and verify that there are no errors showing in cloudera console.

2.4 ActiveMQ (Optional)

ActiveMQ is utilized in the standard installation of LUX. However, the REST alerter can be configured in place of ActiveMQ if preferred. Contact ICG LUX support for configuration details.

1. Download the latest ActiveMQ (e.g. `apache-activemq-<VERSION>.tar.gz`)
2. `tar xf apache-activemq-<VERSION>-bin.tar.gz -C /usr/local/`
3. `cd /usr/local`
4. `sudo ln -s apache-activemq-<VERSION> activemq`
5. **If using CentOS 7/RHEL 7**
    a. Create `/etc/systemd/system/activemq.service` with the content located in Appendix A.2
    b. `sudo chmod 755 /etc/systemd/system/activemq.service`
    c. `sudo systemctl start activemq`
    d. `sudo systemctl enable activemq`
    e. `sudo systemctl status activemq`

6. **If using CentOS 6/RHEL 6**
    a. Create `/etc/init.d/activemq` with the content located in Appendix A.3
    b. `sudo chmod 755 /etc/init.d/activemq`
    c. `sudo chkconfig --add activemq`
    d. `sudo service activemq start`
7. **After** LUX UI installation:
    a. Edit `/usr/local/activemq/conf/activemq.xml` by inserting the contents of Appendix A.4, which adds certs and ssl protocol using `localhost.jks` cert in Tomcat `certs` folder.

3 Prerequisite Software Configuration

3.1 Yum Configuration

1. `sudo yum clean all`
2. `sudo yum check`
3. `sudo yum erase apf`
4. `sudo yum upgrade`

3.1.1 ICG Yum Repository Configuration

If you will be using the online ICG Yum repository to perform the install, follow the steps here in addition to the steps in 3.1

1. `wget https://rpm.icgsolutions.com/pub/icg-repos-2.1-3.noarch.rpm --no-check-certificate`
2. `sudo rpm -Uvh icg-repos-2.1-3.noarch.rpm`
3. `sudo vim /etc/yum.repos.d/icg.repo`
4. Replace `<user>` and `<password>` with the username and password provided to you for repository access.

3.1.2 ICG Repo Direct Download

If you will be downloading the ICG LUX RPMs directly, use the following step:

1. `wget --user <user> --ask-password -r -q -np -A '*.rpm' https://rpm.icgsolutions.com/repo/ --no-check-certificate`
2. Replace `<user>` with the username provided to you by ICG for repository access; and, provide the password when prompted.

3.2 Database Configuration Scripts

If you are using YUM to install, install the DB configuration scripts:

1. `sudo yum install lux-db-config`

If you are not using YUM, install the DB configuration script from RPM:

1. `sudo rpm -Uvh lux-db-config-<VERSION>.noarch.rpm`

Change into the scripts directory to prepare for the next step:

1. `sudo su -`
2. `cd /usr/local/lux/scripts`

3.3 MySQL or Oracle Configuration

MySQL **or** Oracle are required. Follow instructions from section 3.3.1 or 3.3.2 depending on which configuration matches your environment.

3.3.1 MySQL Database

Ensure the port MySQL is listening on (default port 3306) is open for connections through the firewall from the LUX UI host.

1. `mysql -uroot -p`
2. Enter root password assigned at installation
3. `> source prepLuxDBs.sql`
4. `> use luxrule;`
5. `> source rules-mysql.ddl`
6. `> source user_groups.sql`
7. `> quit`

3.3.2 Oracle Database

Ensure the port Oracle is listening on (default port 1521) is open for connections through the firewall from the LUX UI host.

1. Login as the oracle user
2. `export ORACLE_SID=luxrule`
3. `dbca -silent -createDatabase -templateName General_Purpose.dbc -gdbname luxrule -sid luxrule -responseFile NO_VALUE -characterSet AL32UTF8 -memoryPercentage 30 -emConfiguration LOCAL`
4. `sqlplus / as SYSDBA`
5. `> @rules-oracle.ddl`
6. `> quit`

If using PKI as the authentication perform the following steps:

1. Edit `/usr/local/lux/scripts/admin_PKI_setup_oracle.sql` file and add the system admin's CN from their certificate following the directions in the comments in the file.

```
2. sqlplus luxrule/luxrule@luxrule
3. > @admin_PKI_setup_oracle.sql
4. > quit
```

3.4 MongoDB or HBASE Configuration

1. MongoDB **OR** HBASE are required. Follow instructions from section 3.4.1 **or** 3.4.2 depending on which configuration matches your environment.

3.4.1 MongoDB Configuration

Ensure the port MongoDB is listening on default port 27017 is open for connections through the firewall from the LUX UI and LUX Engine hosts (see section 2.3.1).

1. `cd /usr/local/lux/scripts/`
2. `sudo chmod +w setupMongo.js`
3. Edit `setupMongo.js` and change the temporary admin password '<change-me>' (in 2 places) to the admin password you will use.
4. Comment out lines 5 and 6 by adding `//` to the beginning, or remove the lines completely.
5. Change `{"expireAfterSeconds": 432000})` to the value of seconds that the mongo alerts need to persist within the system.
    a. Expiry in 14 days would be a value of `1209600`
6. Save and exit the file
7. Execute the command: `mongo setupMongo.js`

Modify Alert Retention Time from Default 5-day (Optional)

1. Log into Mongo and run the following command, substituting the desired retention time

```
db.runCommand({collMod:'alerts',index:{keyPattern:{createdDate:1},expireAfterSeconds:2592000}})
```

3.4.2 HBASE Configuration

1. `hbase shell`
2. `create 'alert','content'`
3. `create 'aoi','content'`
4. `create 'change','content'`
5. `create 'access','content'`

6. `create 'entity','content'`
7. `create 'entityPosition','content'`

3.4.2.1 SOLR (Optional)

Configuration files for a collection are managed as part of the instance directory. To generate a skeleton of the instance directory run:

a. $ `solrctl instancedir --generate $HOME/solr_configs`
b. go to the generated config set and update it with the attached files, then upload the config set to the server
c. $ `solrctl instancedir --create lux $HOME/solr_configs`
d. check that it was created, lux should be in the returned list
e. $ `solrctl instancedir --list`
f. $ `solrctl collection --create access -s 1 -c lux`
g. $ `solrctl collection --create alert -s 1 -c lux`
h. $ `solrctl collection --create change -s 1 -c lux`
i. $ `solrctl collection --create entityPosition -s 1 -c lux`
j. $ `solrctl collection --create aoi -s 1 -c lux`
k. $ `solrctl collection --create entity -s 1 -c lux`

3.4.2.2 Security Unsecured

1. Within HBASE Settings:
    a. hbase.url=192.168.5.131
    b. hbase.kerberos=false
    c. hbase.user=
    d. hbase.keytab=
    e. solr.url=http://192.168.5.131:8983/solr/

Build with your normal profiles + hbase

3.4.3 Tesseract-ocr Configuration (optional)

1. sudo yum --enablerepo epel-testing install tesseract.x86_64 tesseract-langpack-eng.noarch

4 LUX Installation

4.1 User Interface (UI)

Use YUM to install the LUX UI if your environment is not on a disconnected LAN (see 4.1.2). If your environment is not able to access the internet, or ICG YUM repositories, use the RPMs provided in your LUX distribution software package (see 4.1.1).

The LUX UI installation includes the installation of tomcat.

4.1.1 Install LUX packages from RPM

If you are **not** using YUM to install the LUX UI, you will need to install JSVC manually. If you are using YUM to install the LUX UI, skip to 4.1.2.

1. Install JSVC
    a. `sudo yum install jsvc`
    b. If this does not run, try the following:
        i. On Centos 6.x: `sudo rpm -Uvh jakarta-commons-daemon-jsvc-1.0.1-8.9.el6.x86_64.rpm`
        ii. On Centos 7.x: `sudo rpm -Uvh apache-commons-daemon-jsvc-1.0.15-11.fc24.x86_64.rpm`
2. `sudo rpm -Uvh lux-ui-<VERSION>-noarch.rpm`

4.1.2 Install LUX package from YUM

Depending on your install version, use one of the following commands:

1. `sudo yum install lux-ui`
2. `sudo yum install lux-ui-jblocks`

4.1.3 Configure LUX UI

1. Edit `/etc/init.d/lux`
    a. Update the variable `JAVA_HOME` with the path to java 1.8 install.
    b. If using REST instead of ActiveMQ, remove the line that specifies `-Djavax.net.ssl.trustStorePassword`
2. Change ownership of the LUX UI directory to the 'LUX' user:
    a. `chown -R lux:lux /usr/local/lux/ui/`
3. Within: `/usr/local/lux/ui/webapps/lux/WEB-INF/classes/spring-context.properties` configure the following required options:

i. `jms.server.url` - URL and connection type of the JMS server you are using. If you are following this guide to setup your JMS server, you can leave default.
ii. `jms.user` - Username to connect to your JMS server. If you are using this guide, leave default.
iii. `jms.password` - Password for the user. If you are using this guide, leave default.
iv. `lux.webapp.url` - The LUX server URL. This should be set to your server's URL (i.e. `https://example.com/lux`)
v. `builder.run` - If `=true` then it enables the LUX UI Watchboard functionality and rule folder warning colorations. Set to (default) `=false` if watchboard functionality is not desired.

b. The lux rpm will install the forms, cop, store, template and mapserver files in `/usr/local/lux/ui/<forms,cop,stores,templates,mapserver>` but will not override existing files modified by the end-user.
c. The lux.war file also installs the same files to `/usr/local/lux/ui/webapps/lux<forms,cop,stores,templates,mapserver>`. If the release has updated the contents of these folders, any differences between the end-user's version and the lux.war installed version has to be justified and rectified.
d. Any instance of `localhost` in `spring-context.properties` should be replaced with the actual server ip address or DNS host name or the server won't serve remotely. There should be multiple occurrences that need to be replaced.
e. Edit: `/usr/local/lux/ui/webapps/lux/WEB-INF/classes/liveAlertsKmlTemplate.vm` and replace `localhost` with the hostname of the server.
f. Edit: `/usr/local/lux/ui/webapps/lux/WEB-INF/classes/spring-security.properties` and replace `localhost` with the hostname of the server.

4. If using MongoDB:
   a. Configure settings in `/usr/local/lux/ui/webapps/lux/WEB-INF/classes/mongo.properties` by changing `mongo.url` and `mongo.entity.url` to equal the hostname of your server.
4. If using HBASE:
   a. Configure settings in `/usr/local/lux/ui/webapps/lux/WEB-INF/classes/hbase.properties`
5. Configure MySQL connection settings in `/usr/local/lux/ui/webapps/lux/WEB-INF/classes/db-rules.properties`
   a. Change any instances of `localhost` to equal the hostname of your server.
6. If using the rest alerter with the LUX Engine:

a. Replace the content of `/usr/local/lux/ui/webapps/lux/WEB-INF/classes/ingest.properties` with:
   i. `ingest.alert.rest.enabled=true`
b. Comment out all instances of 'jms' within `/usr/local/lux/ui/webapps/lux/WEB-INF/classes/spring-context.properties`
c. Delete `/usr/local/lux/ui/webapps/lux/WEB-INF/spring/jms.xml`

7. Add Installation details, such as customer name, LUX serial number, and license expiry dates to `/usr/local/lux/ui/webapps/lux/WEB-INF/classes/license.properties`
8. Add UI Installation name and validate product version in: `/usr/local/lux/ui/webapps/lux/WEB-INF/classes/version.properties`
9. Add the LUX Engine Version to: `/usr/local/lux/ui/webapps/lux/WEB-INF/classes/engine.properties`
10. `sudo service lux start`
11. `sudo chkconfig lux on`
12. `cd /usr/local/lux/ui/logs; sudo tail -f catalina.out` (and check for errors, e.g., `/var/tmp/geowebcache is not writable directory.` Rectify and restart lux.)
13. Navigate to the LUX UI Webpage
    a. If following this guide, the URL and login credentials will be as follows:
    b. URL: https://<HOSTNAME>/lux/
    c. Default credentials:
       i. Username - admin
       ii. Password - admin

4.1.4 Configure Timeline (Optional)

1. Modify the following file with the desired timeline values for the UI:

`/usr/local/lux/ui/webapps/lux/config/timeline/timerangeselector.json`

4.2 LUX Engine Installation

Either install the engine from RPM files provided from ICG using the instructions in section 4.2.1, or install directly from ICG's YUM repository using the instructions in 4.2.2. You may need to execute the following instructions with sudo or as root.

4.2.1 Install packages from RPM

1. `sudo rpm -Uvh --replacefiles lux-engine-<VERSION>.noarch.rpm`
2. Copy a valid `lux.lic` license file provided by ICG to the `/usr/local/lux/engine/license` directory.
3. Copy `/usr/local/lux/engine/AdminConsole.war` to `/usr/local/lux/ui/webapps` on the UI node

4. Edit `/etc/init.d/luxengine` replace the value of `JAVA_HOME` with the location of your java root directory
5. `chkconfig luxengine on`

4.2.2 Install packages from Yum

1. Depending on your install version, use one of the following commands:
    a. `sudo yum install lux-engine`
    b. `sudo yum install lux-engine-jblocks`
2. Copy a valid `lux.lic` license file provided by ICG to the `/usr/local/lux/engine/license` directory.
3. Copy `/usr/local/lux/engine/AdminConsole.war` to `/usr/local/lux/ui/webapps` on the UI node
4. Edit `/etc/init.d/luxengine` replace the value of `JAVA_HOME` with the location of your java root directory
5. `chkconfig luxengine on`

4.2.3 Basic LUX Engine Configuration

1. Edit `/usr/local/lux/engine/EngineMain/data/conf/lux.properties` change all occurrences of `localhost` to the hostname of the LUX UI server.
2. Edit `/usr/local/lux/engine/EngineMain/data/conf/engine.properties` change the value of the `admin.console.rest.url` to the hostname of the LUX UI server.
3. Edit `/usr/local/lux/engine/EngineMain/data/conf/jms.xml` change all occurrences of `localhost` to the hostname of the LUX UI server.

4.2.4 Starting and stopping LUX Engine

1. To start LUX Engine: `sudo service luxengine start`
2. To stop LUX Engine: `sudo service luxengine stop`

4.2.4 Install Admin Console

1. If using the LUX UI, verify that the Admin Console .war file has been copied from the engine host to the UI host (See section 4.2.2).
2. If using the Engine as standalone, install Apache Tomcat on the Engine host and copy the `/usr/local/lux/engine/AdminConsole.war` file to the `$CATALINA_HOME/webapps/` directory.
3. Login to the LUX Engine Administration Console
    a. If following this guide, the URL and login credentials will be as follows:
    b. URL: https://<HOSTNAME>/AdminConsole/
    c. Default credentials:

i. Username - admin
ii. Password - admin1

Appendix A

A.1 Mongod Configuration File Contents

1. Within `/etc/init.d/mongod` :

```
#!/bin/bash

### BEGIN INIT INFO
# Provides:        mongod
# Required-Start:  $network $local_fs $remote_fs
# Required-Stop:   $network $local_fs $remote_fs
# Should-Start:    $named
# Should-Stop:
# Default-Start:   2 3 4 5
# Default-Stop:    0 1 6
# Short-Description: An object/document-oriented database
# Description:     MongoDB is a high-performance, open source, schema-free
#                  document-oriented data store that's easy to deploy, manage
#                  and use. It's network accessible, written in C++ and offers
#                  the following features:
#
#                  * Collection oriented storage - easy storage of object-
#                        style data
#                        * Full index support, including on inner objects
#                        * Query profiling
#                        * Replication and fail-over support
#                        * Efficient storage of binary data including large
#                        objects (e.g. videos)
#                        * Automatic partitioning for cloud-level scalability
#
#                  High performance, scalability, and reasonable depth of
#                  functionality are the goals for the project.
### END INIT INFO

DAEMON=/usr/local/mongodb/bin/mongod
NAME=mongo
RUNUSER=mongo
PIDFILE=/var/run/${NAME}.pid
CONF=/etc/mongod.conf

if [ -f /etc/default/$NAME ] ; then
      . /etc/default/$NAME
fi
```

```
. /etc/init.d/functions

case "$1" in
  start)
            # Recommended ulimit values for mongod or mongos
            # See
http://docs.mongodb.org/manual/reference/ulimit/#recommended-settings
            #
            ulimit -f unlimited
            ulimit -t unlimited
            ulimit -v unlimited
            ulimit -n 64000
            ulimit -m unlimited

            # In dash, ulimit takes -p for maximum user processes
            # In bash, it's -u
            if readlink /proc/$$/exe | grep -q dash
            then
                  ulimit -p 64000
            else
                  ulimit -u 64000
      fi

      daemon --pidfile="$PIDFILE" --user="$RUNUSER" "$DAEMON &"
      echo `ps -ef | grep mongod | grep ^mongo | grep -v bash | awk '{print
$2}'`  > $PIDFILE ;;
  stop)
      killproc -p $PIDFILE $DAEMON ;;
  status)
      status -p $PIDFILE $DAEMON ;;
  restart)
      stop && start ;;
esac
```

A.2 ActiveMQ Configuration File Contents

1. Copy the following data into `/etc/systemd/system/activemq.service` :

```
[Unit]
Description=ActiveMQ service
After=network.target

[Service]
Type=forking
ExecStart=/usr/local/activemq/bin/activemq start
```

```
ExecStop=/usr/local/activemq/bin/activemq stop
User=root
Group=root
Restart=always
RestartSec=9
StandardOutput=syslog
StandardError=syslog
SyslogIdentifier=activemq

[Install]
WantedBy=multi-user.target
```

A.3 ActiveMQ Configuration File Contents

1. Copy the following data into `/etc/init.d/activemq` :

```
#!/bin/bash
#
# activemq        #Starts ActiveMQ.
#
#
# chkconfig: 345 70 12
# description: ActiveMQ is a JMS Messaging Queue Server.
### BEGIN INIT INFO
# Provides: $activemq
### END INIT INFO
ACTIVEMQ_HOME=/usr/local/activemq
ACTIVEMQ_SCRIPT=$ACTIVEMQ_HOME/bin/activemq
# Source function library.
. /etc/init.d/functions
RETVAL=0
umask 077
start() {
      echo "Deleting old messages from /extradrive/activeMQ/data/kahadb"
      rm -rf /usr/local/activemq/data/kahadb/*
      echo -n $"Starting ActiveMQ: "
      daemon $ACTIVEMQ_SCRIPT start
      echo
      return $RETVAL
}
stop() {
      echo -n $"Shutting down ActiveMQ: "
```

```
        daemon "$ACTIVEMQ_SCRIPT stop"
        echo
        return $RETVAL
}
restart() {
        stop
        start
}
case "$1" in
 start)
        start
        ;;
 stop)
        stop
        ;;
 restart|reload)
        restart
        ;;
 *)
        echo $"Usage: $0 {start|stop|restart}"
        exit 1
esac
exit $?
```

A.4 ActiveMQ.XML File Contents

1. Edit `/usr/local/activemq/conf/activemq.xml` by inserting the content below, beginning within the `<broker/>` tag, just before the `<transportConnectors/>` section:

```
    <sslContext>
                <!-- Note: that the keystore value must be an absolute URL, not
a filename -->
              <sslContext
                  keyStore="file:///usr/local/lux/ui/certs/keystore.jks"
                  keyStorePassword="changeit"
                  trustStore="file:///usr/local/lux/ui/certs/truststore.jks"
                  trustStorePassword="changeit"/>
          </sslContext>
```

Also edit the lines in activemq.xml within `<transportConnections/>` so they look like this:

```
    <transportConnectors>
```

```
            <transportConnector name="ssl"
uri="ssl://0.0.0.0:61616?maximumConnections=1000&amp;wireFormat.maxFrameSize=1
04857600"/>
        </transportConnectors>
```