

LUX Installation Guide

Updated: May 2017

1 Prerequisites

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	Version	Download URL
MySQL	5.3+	https://www.mysql.com/downloads
MongoDB	3.4.4	https://www.mongodb.com/download-center#community
Oracle JDK	1.8	http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
ActiveMQ	5.3+	http://activemq.apache.org/download.html

2 Prerequisite Software Installation

2.1 JDK 1.8

1. Download or obtain a JDK 1.8 rpm (e.g. `jdk-8u121-linux-x64.rpm`).
2. `sudo rpm -Uvh jdk-8u121-linux-x64.rpm`
3. `sudo /usr/sbin/alternatives --install "/usr/bin/java" "java" "/usr/java/default/bin/java" 2`
4. `sudo /usr/sbin/alternatives --config java`
5. Choose the 1.8 alternative

2.2 MYSQL 5.7

1. Remove mariadb if installed: `sudo rpm -e --nodeps mariadb-libs-1:5.5.41-2.el7_0.x86_64`
2. For CentOS 7.x: download `mysql-5.7.14-1.el7.x86_64.rpm-bundle.tar`
For CentOS 6.x: download `mysql-5.7.14-1.el6.x86_64.rpm-bundle.tar`
3. `tar xf mysql-5.7.14-1.el7.x86_64.rpm-bundle.tar`
4. `sudo rpm -Uvh mysql-community-common-5.7.14-1.el7.x86_64.rpm` (depending on your system configuration, you may need to use the `--force` or `--nodeps` options for commands 4-7)
5. `sudo rpm -Uvh mysql-community-libs-5.7.14-1.el7.x86_64.rpm`
6. `sudo rpm -Uvh mysql-community-client-5.7.14-1.el7.x86_64.rpm`
7. `sudo rpm -Uvh mysql-community-server-5.7.14-1.el7.x86_64.rpm`
8. `sudo semanage fcontext --add --type mysqld_db_t '/var/lib/mysql'` (if `semanage` is not available, `sudo yum install policycoreutils-python`)
9. `sudo restorecon -r /var/lib/mysql`
10. `sudo chmod -R 755 /var/lib/mysql`
11. `sudo chown -R mysql:mysql /var/lib/mysql`
12. `sudo mysqld --initialize`
13. `sudo chown -R mysql:mysql /var/lib/mysql`
14. `sudo service mysqld start`
15. `sudo chkconfig mysqld on`
16. `sudo grep 'temporary password' /var/log/mysqld.log`
17. `mysql_secure_installation`
18. Enter the temporary password from above, and change the password. By default password validation is enabled, your root password must meet the minimum complexity described below:

Variable Name	Value
<code>validate_password_check_user_name</code>	off
<code>validate_password_dictionary_file</code>	
<code>validate_password_length</code>	8
<code>validate_password_mixed_case_count</code>	1
<code>validate_password_number_count</code>	1
<code>validate_password_policy</code>	MEDIUM
<code>validate_password_special_char_count</code>	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. Login to the mysql root account and issue the following commands:

```
UNINSTALL PLUGIN validate_password;
UPDATE mysql.user SET authentication_string = PASSWORD('MyNewPass')
    WHERE User = 'root' AND Host = 'localhost';
FLUSH PRIVILEGES;
```

2.3.1 MongoDB 3.4.4

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 Tarball

1. For CentOS 6.x: download `mongodb-linux-x86_64-rhel62-3.4.4.tgz`
For CentOS 7.x: download `mongodb-linux-x86_64-rhel70-3.4.4.tgz`
2. `tar -zxvf mongodb-linux-x86_64-rhel70-3.4.4.tgz`
3. `sudo cp -R -n mongodb-linux-x86_64-rhel70-3.4.4/ /usr/local`
4. `sudo ln -s /usr/local/mongodb-linux-x86_64-rhel70-3.4.4
/usr/local/mongodb`
5. Create the file `/etc/profile.d/mongodb.sh` and insert `export
PATH=$PATH:/usr/local/mongodb/bin` as the first line.
6. `./etc/profile`
7. Create a data folder `sudo mkdir -p /data/db`
8. `sudo useradd -c "MongoDB Service" -d /usr/local/mongodb -M -r -U mongo`
9. `sudo chown -R mongo:mongo /usr/local/mongodb/*`
10. `sudo chown -R mongo:mongo /data/*`
11. If using RHEL/CentOS 7: 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
```

- a. `sudo systemctl daemon-reload`
 - b. Start MongoDB using `sudo systemctl start mongod`
12. If using RHEL/CentOS 6: Create the file `/etc/init.d/mongod` with the content from appendix A.
- a. `sudo chmod 755 /etc/init.d/mongod`
 - b. `sudo chkconfig --add mongod`
 - c. Start MongoDB using `sudo service mongod start`

From Yum

1. Create `/etc/yum.repos.d/mongodb-org-3.2.repo` with the following content:

```
[mongodb-org-3.2]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/3.4/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-3.4.asc
```
2. `sudo yum install -y mongodb-org`
3. `sudo semanage port -a -t mongod_port_t -p tcp 27017`
4. `sudo service mongod start`
5. `sudo chkconfig mongod on`

Check Mongo configuration to ensure mongod is listening on all interfaces

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 `bind_ip=0.0.0.0` so that mongod does indeed listen on all interfaces.

2.3.2 HBASE

1. Download quick start vm 5.5
2. Run the launch cloudera enterprise trial
3. Login to web console
4. (there are issues with time, so make sure the vm time matches your system time, might need to start `ntpd` or something)
5. Restart the cluster until there are no errors showing in cloudera console

2.4 ActiveMQ

1. Download the latest ActiveMQ (e.g. `apache-activemq-5.14.4-bin.tar.gz`)

2. `tar xf apache-activemq-5.14.4-bin.tar.gz`
3. `mv apache-activemq-5.14.4 /usr/local/`
4. `cd /usr/local`
5. `sudo ln -s apache-activemq-5.14.4 activemq`
6. Create `/etc/init.d/activemq` with the following content:

```
#!/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 $?

```

7. `sudo chmod 755 /etc/init.d/activemq`
8. `sudo chkconfig --add activemq`
9. (After LUX UI installation) Edit `/usr/local/activemq/conf/activemq.xml` and add certs and ssl protocol using `localhost.jks` cert in Tomcat `certs` folder. Add the following lines to `activemq.xml`, 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&wireFormat.maxFrameSize=104857600"/>
</transportConnectors>

```

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.

3.2 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-2.4.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 Configuration

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`

3.4 MongoDB or HBASE

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

1. Edit `setupMongo.js` and change the temporary admin password '<changeme>' (in 2 places) to the admin password you will use.
2. `mongo setupMongo.js`

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'`

3.4.3 SOLR

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`

8. Security

Unsecured

1. In
2. `hbase.url=192.168.5.131`
3. `hbase.kerberos=false`
4. `hbase.user=`
5. `hbase.keytab=`
6. `solr.url=http://192.168.5.131:8983/solr/`

Build with your normal profiles + hbase

3.6 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. If your environment is not able to access the internet, or ICG YUM repositories, use the RPMs provided in your LUX distribution software package.

The LUX UI installation includes the installation of tomcat.

4.1.1 Install 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` If this does not run, then try this:
 - i. On Centos 6.X: `sudo rpm -Uvh
jakarta-commons-daemon-jsvc-1.0.1-8.9.e16.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-2.x-SNAPSHOT.noarch.rpm`

4.1.2 Install packages from YUM

1. `sudo yum install lux-ui`

4.1.3 Configure LUX UI

1. Edit `/etc/init.d/lux` and update the variable `JAVA_HOME` with the path to java 1.8 install.
2. Start the UI so that tomcat deploys lux.war, and then stop the UI so you can configure tomcat. *If LUX (e.g., tomcat) does not start in step a below (i.e., if you do not see tomcat running in step b), then check that the server name resolves in DNS. If it does not, then add an entry in `/etc/hosts` that names the host and identifies the host's IP address. Tomcat won't start unless it can find the host's ip address from the hostname. Do not proceed beyond step a if tomcat is not running. (If you start tomcat as root, or any user other than as lux, you will have to chown all the files that tomcat creates deploying lux.war.)*
 - a. `sudo service lux start`
 - b. `ps auxw | grep tomcat` (visually check that tomcat is running)
 - c. `cd /usr/local/lux/ui/logs`
 - d. `sudo tail -f catalina.out` (and wait for output to stop, "server startup")
 - e. `ctrl-C` out of tail
 - f. `sudo service lux stop`
3. Now, tomcat has expanded the file `/usr/local/lux/ui/webapps/lux.war` to the lux folder in that same parent folder. In `/usr/local/lux/ui/webapps/lux/WEB-INF/classes/spring-context.properties` configure the following required options:
 - a. `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.
 - b. `jms.user` - Username to connect to your JMS server. If you are using this guide, leave default.
 - c. `jms.password` - Password for the user. If you are using this guide, leave default.
 - d. OrientDB connection info (required only if using the Entity Manager)
 - e. `lux.webapp.url` - The LUX server URL. This should be set to your server's URL (i.e. `https://example.com/lux`)
 - f. 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.
 - g. 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.

- h. 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 9 occurrences that need to be replaced.
2. Configure Mongo settings in
`/usr/local/lux/ui/webapps/lux/WEB-INF/classes/mongo.properties`
OR
Configure HBASE settings in
`/usr/local/lux/ui/webapps/lux/WEB-INF/classes/hbase.properties`
3. Configure MySQL connection settings in
`/usr/local/lux/ui/webapps/lux/WEB-INF/classes/db-rules.properties`
4. `sudo rm -rf /usr/local/lux/ui/webapps/examples`
5. `sudo service lux start`
6. `sudo chkconfig lux on`
7. `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.)

4.2 Engine

4.2.1 Install packages from RPM

1. `sudo rpm -Uvh --replacefiles lux-engine-2.4.noarch.rpm`
2. Copy a valid **lux.lic** license file provided by ICG to the `/usr/local/lux/engine/license` folder
3. Copy `/usr/local/lux/AdminConsole.war` to `/usr/local/lux/ui/webapps` on the UI node
4. Update JMS URL in `LUXEngine/EngineMain/data/conf/jms.xml`
5. Update AdminConsole URL in
`LUXEngine/EngineMain/data/conf/engine.properties`
6. Update UI URLs in `LUXEngine/EngineMain/data/conf/lux.properties`
7. `LUXEngine/bin/start-lite.sh`

4.2.2 Install packages from Yum

1. `sudo yum install lux-engine`

4.2.3 Install Admin Console

If using the LUX UI, copy the `/usr/local/lux/AdminConsole.war` file from the engine host to the UI host in the `/usr/local/lux/ui/webapps/` directory.

If using the Engine as standalone, install Apache Tomcat on the Engine host and copy the `/usr/local/lux/AdminConsole.war` file to the `$CATALINA_HOME/webapps/` directory.

Appendix A

/etc/init.d/mongod file contents

```
#!/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

```