JobScheduler - Amazon AMI Installation

March 2015

Contact Information

Software- und Organisations-Service GmbH

Giesebrechtstr. 15
D-10629 Berlin
Germany

Telephone +49 (0)30 86 47 90-0
Telefax +49 (0)30 8 61 33 35
Mail info@sos-berlin.com
Web http://www.sos-berlin.com

Last Updated: 03/13/2015 12:00 PM

This documentation is based on JobScheduler Version 1.7.4169.

We would appreciate any feedback you have, or suggestions for changes and improvements; please forward your comments to info@sos-berlin.com.

Table of Contents

1 Introduction

This document describes installation of the **JobScheduler** together with a MySQL™ database on an Amazon AMI Linux 64Bit image.

Please note: the Amazon AMI image is "headless", meaning that it does not come with the graphic libraries necessary to run **JOE (JobScheduler Object Editor)** or the **JID (JobScheduler Information Dashboard)**. See "JOE and JID" chapter (page 9) of this document if you want to use these **JobScheduler** features in your image and/or find more information.

An Amazon Web Services (AWS) account is required so that you can access and use the Amazon AMI image. See http://aws.amazon.com for more information about AWS and registering for an AWS account.

Log into your AWS account.

Open the *EC2* tab of the AWS Management Console to create and launch an instance of an Amazon AMI Linux 64Bit image. See the AWS GettingStartedGuide for detailed information about this procedure.

Once you have launched the Amazon AMI Linux 64Bit instance, you will find it in the *Instances* list on the *EC2* tab of the AWS Management Console. Note that to connect the instance via a SSH Client and to install and use the **JobScheduler** on this instance, you may need specific information such as the *Public DNS Name* of the instance. You can obtain such information by clicking on the instance in the *Instances* list.

You must edit the security group (default = quicklaunch-1) of your instance. To do this go to *Security Groups* in the *EC2* tab of the AWS Management Console. Add rules for the MySQL™ port 3306 and the **JobScheduler** port 4444.

Now you can connect to the Amazon AMI Linux 64Bit instance via SSH as user `ec2-user` either with a standalone SSH Client or from your browser using the Java SSH Client. Select your Amazon AMI Linux 64Bit instance in the *Instances* list and click on *Instance Actions -> Connect* to get more information.
If you use PuTTy as a standalone SSH Client, you need to convert the private key file (pem) into a ppk file (see http://docs.amazonwebservices.com/AWSEC2/latest/UserGuide/putty.html).

If you need root permission on your Amazon AMI Linux instance then call

```
sudo ...
```

2 Installation of MySQL

Install MySQL™ from the repositories.

```
sudo yum groupinstall "MySQL Database"
```

Start the MySQL™ server.

```
sudo service mysqld start
```

2.1 Create a MySQL database

Login as MySQL™ root user.

```
mysql -u root
```

Create a database `scheduler` for the user `scheduler` with `scheduler` as the password.

```
mysql> create database scheduler;
mysql> grant all on scheduler.* to 'scheduler'@'localhost' identified by 'scheduler';
mysql> exit;
```

Test the login with the MySQL™ `scheduler` user.

```
mysql -u scheduler -pscheduler scheduler

mysql> exit;
```

2.2 Download MySQL JDBC Driver

Note that the exact download link for the MySQL JDBC driver may change: See http://dev.mysql.com/downloads/connector/j/ to get the current url.

```
cd /tmp
wget http://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-5.1.20.tar.gz
     /from/ftp://ftp.fu-berlin.de/unix/databases/mysql/
gzip -d mysql-connector-java-5.1.20.tar.gz
tar -xf mysql-connector-java-5.1.20.tar
```

3 Installation of Oracle Java JRE 32Bit

The `wget` command doesn't work here, because Oracle™ wants a confirmation of the licence agreement before the download takes place. Download i.e. `jre-6u26-linux-i586.bin` direct from the Java SE Downloads - Oracle site and transfer it to the `/tmp` directory of the Amazon AMI Linux 64Bit instance via SFTP.

Install it i.e. at `/opt/jre6/jre1.6.0_26`

```
cd /tmp
chmod a+x jre-6u26-linux-i586.bin
sudo mkdir /opt/jre6
sudo mv jre-6u26-linux-i586.bin /opt/jre6
cd /opt/jre6
sudo ./jre-6u26-linux-i586.bin
```

4 Installation of the JobScheduler

4.1 Download the JobScheduler

Download the current **JobScheduler** from Sourceforge. The example below will download release 1.3.12.2137.

```
cd /tmp
wget http://sourceforge.net/projects/jobscheduler/files/scheduler_linux.1.3.12.2137.tar.gz/download
gzip -d scheduler_linux.1.3.12.2137.tar.gz
tar -xf scheduler_linux.1.3.12.2137.tar
```

4.2 Configure and start the JobScheduler setup

Edit the `/tmp/jobscheduler.1.3.12.2137/scheduler_install.xml`, which is listed below, as follows:

- Set both installation paths (*installpath*, *UserPathPanelElement*) to `/home/ec2-user/`.
- Set the above database connection as specified in "Create a MySQL Database" section (page 5) above.
- Set *launchScheduler* to `no`, because you need to change the Java™ environment (page 9) after the installation but before the **JobScheduler** starts. Otherwise the **JobScheduler** won't find the correct Java™ VM.
- In *schedulerHost* type the *Public DNS Name* of the Amazon AMI Linux 64Bit instance (see the AWS Management Console (page 4))

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<AutomatedInstallation langpack="eng">
    <com.izforge.izpack.panels.UserInputPanel id="home">
        <userInput/>
    </com.izforge.izpack.panels.UserInputPanel>
    <com.izforge.izpack.panels.UserInputPanel id="licences">
        <userInput>
            <entry key="licenceOptions" value="GPL"/>
            <entry key="licence" value=""/>
        </userInput>
    </com.izforge.izpack.panels.UserInputPanel>
    <com.izforge.izpack.panels.HTMLLicencePanel id="gpl_licence"/>
    <com.izforge.izpack.panels.HTMLLicencePanel id="commercial_licence"/>


    <!-- installation paths -->
    <com.izforge.izpack.panels.TargetPanel id="target">
        <installpath>/home/ec2-user</installpath>
    </com.izforge.izpack.panels.TargetPanel>
    <com.izforge.izpack.panels.UserPathPanel id="userpath">
        <UserPathPanelElement>/home/ec2-user</UserPathPanelElement>
    </com.izforge.izpack.panels.UserPathPanel>


    <com.izforge.izpack.panels.PacksPanel id="package">
        <pack index="0" name="Job Scheduler" selected="true"/>
        <pack index="1" name="Update Service" selected="false"/>
        <pack index="2" name="Database Support" selected="true"/>
        <pack index="3" name="Web" selected="false"/>
        <pack index="4" name="Managed Jobs" selected="false"/>
        <pack index="5" name="Housekeeping Jobs" selected="true"/>
        <pack index="6" name="MySQL" selected="false"/>
        <pack index="7" name="Cron" selected="false"/>
    </com.izforge.izpack.panels.PacksPanel>
    <com.izforge.izpack.panels.UserInputPanel id="network">
        <userInput>
```

```
            <entry key="schedulerHost" value="localhost"/>
            <entry key="schedulerPort" value="4444"/>
            <entry key="schedulerId" value="scheduler"/>
            <entry key="schedulerAllowedHost" value="0.0.0.0"/>


            <!-- launch JobScheduler after installation -->
            <entry key="launchScheduler" value="no"/>



        </userInput>
    </com.izforge.izpack.panels.UserInputPanel>
    <com.izforge.izpack.panels.UserInputPanel id="cluster">
        <userInput>
            <entry key="clusterOptions" value=""/>
        </userInput>
    </com.izforge.izpack.panels.UserInputPanel>
    <com.izforge.izpack.panels.UserInputPanel id="smtp">
        <userInput>
            <entry key="mailServer" value="localhost"/>
            <entry key="mailPort" value="25"/>
            <entry key="smtpAccount" value=""/>
            <entry key="smtpPass" value=""/>
            <entry key="mailFrom" value=""/>
            <entry key="mailTo" value=""/>
            <entry key="mailCc" value=""/>
            <entry key="mailBcc" value=""/>
        </userInput>
    </com.izforge.izpack.panels.UserInputPanel>
    <com.izforge.izpack.panels.UserInputPanel id="email">
        <userInput>
            <entry key="mailOnError" value="yes"/>
            <entry key="mailOnWarning" value="yes"/>
            <entry key="mailOnSuccess" value="no"/>
            <entry key="jobEvents" value="off"/>
        </userInput>
    </com.izforge.izpack.panels.UserInputPanel>
    <com.izforge.izpack.panels.UserInputPanel id="update">
        <userInput>
            <entry key="checkForUpdateStarttime" value="20:00"/>
            <entry key="checkForUpdateStartday" value="1"/>
            <entry key="autoUpdateDownload" value="0"/>
        </userInput>
    </com.izforge.izpack.panels.UserInputPanel>


    <!-- database connection data -->
    <com.izforge.izpack.panels.UserInputPanel id="database">
        <userInput>
            <entry key="databaseDbms" value="mysql"/>
            <entry key="databaseCreate" value="on"/>
        </userInput>
    </com.izforge.izpack.panels.UserInputPanel>
    <com.izforge.izpack.panels.UserInputPanel id="dbconnection">
        <userInput>
            <entry key="databaseHost" value="localhost"/>
            <entry key="databasePort" value="3306"/>
            <entry key="databaseSchema" value="scheduler"/>
            <entry key="databaseUser" value="scheduler"/>
            <entry key="databasePassword" value="scheduler"/>
            <entry key="connectorJTDS" value="yes"/>
        </userInput>
    </com.izforge.izpack.panels.UserInputPanel>
    <com.izforge.izpack.panels.UserInputPanel id="jdbc">
        <userInput>
            <entry key="connector" value="/tmp/mysql-connector-java-5.1.20/mysql-connector-java-5.1.20.jar"/>
        </userInput>
    </com.izforge.izpack.panels.UserInputPanel>



    <com.izforge.izpack.panels.UserInputPanel id="cron">
        <userInput>
            <entry key="cronCrontab" value="/etc/crontab"/>
            <entry key="cronSystab" value="1"/>
            <entry key="cronTimeout" value="600"/>
            <entry key="cronChangeUser" value=""/>
            <entry key="cronChangeCommand" value=""/>
```

```
        </userInput>
    </com.izforge.izpack.panels.UserInputPanel>
    <com.izforge.izpack.panels.InstallPanel id="install"/>
    <com.izforge.izpack.panels.ProcessPanel id="process"/>
    <com.izforge.izpack.panels.FinishPanel id="finish"/>
</AutomatedInstallation>
```

Example: scheduler_install.xml

Start the setup (without root privileges and specifying the scheduler_install.xml file to avoid X server problems) using:

```
cd /tmp/jobscheduler.1.3.12.2137/
./setup.sh -u scheduler_install.xml
```

4.3 Set JAVA_HOME

Edit the file `~/scheduler/bin/jobscheduler_environment_variables.sh`, setting the `JAVA_HOME` environment variable to the value set when configuring the Oracle™ JRE 32Bit installation (page 6).

```
...
#test -z "$JAVA_HOME" && JAVA_HOME="/usr/lib/jvm/java-1.6.0-openjdk-1.6.0.0.x86_64/jre"
JAVA_HOME="/opt/jre6/jre1.6.0_26"
...
```

Now you can start the **JobScheduler**.

```
~/scheduler/bin/jobscheduler.sh start
```

and open **JOC (JobScheduler Operations Center)** via `http://[Public DNS of instance]:4444`

4.4 Perl

If you want to use the **JobScheduler** internal API with Perl then you need the 32Bit runtime libraries of Perl.

Type the following to install these libraries.

```
sudo yum install perl-libs.i686
```

Now the Perl 32Bit runtime libraries are stored at `/usr/lib/perl5/5.10.0/i386-linux-thread-multi/CORE/libperl.so` (the path may change). Create a symlink in `~/scheduler/lib` so that the **JobScheduler** find the Perl 32Bit environment.

```
cd ~/scheduler/lib/
ln -s /usr/lib/perl5/5.10.0/i386-linux-thread-multi/CORE/libperl.so libperl.so
```

4.5 JOE and JID

JOE (JobScheduler Object Editor) and the **JID (JobScheduler Information Dashboard)** use SWT, which requires libraries that are not included in the Amazon AMI image, because it is headless. You must install an X-Windows system and the GTK. Then you can open **JOE** and **JID** with a local X-Server (e.g. Xming on Windows) if you have configured the SSH connection with an X forwarding.

The performance when using **JOE** and **JID** with an X forwarding and a local X-Server can be very slow. We recommend that you install **JOE** and **JID** local. **JOE** has a inside to transfer the configuration to the Amazon AMI instance. For **JID** you must make sure that it can connect to the **JobScheduler** database.

If you want to install an X-Windows system and the GTK then we change the SWT. After the installation of the **JobScheduler** you have a 32Bit SWT in the `~/scheduler/lib/` directory. The 32Bit SWT miss more libraries than the 64Bt SWT, so we will use a 64Bit SWT. Follow the following steps:

4.5.1 Install 64Bit SWT

You get download links at http://download.eclipse.org/eclipse/downloads/drops/R-3.7.2-201202080800/. I have used http://mirror.netcologne.de/eclipse//eclipse/downloads/drops/R-3.7.2-201202080800/swt-3.7.2-gtk-linux-x86_64.zip.

```
cd /tmp
mkdir swt
cd swt
wget http://mirror.netcologne.de/eclipse//eclipse/downloads/drops/R-3.7.2-201202080800/swt-3.7.2-gtk-linux-x86_64.zip
unzip swt-3.7.2-gtk-linux-x86_64.zip
rm ~/scheduler/lib/swt.jar
cp swt.jar ~/scheduler/lib
```

Edit the startscript of **JOE** (`~/scheduler/bin/jobeditor.sh`) and **JID** (`~/scheduler/bin/dashboard.sh`). JAVA_HOME is pointed to the above 32Bit Oracle Java JRE but now we need a 64Bit JRE.

```
...
#test -z "$JAVA_HOME" && JAVA_HOME="/usr/lib/jvm/java-1.6.0-openjdk-1.6.0.0.x86_64/jre"
JAVA_HOME="/usr/lib/jvm/java-1.6.0-openjdk-1.6.0.0.x86_64/jre"
...
```

4.5.2 Install the X-Windows system

Install the X-Windows system with `yum`.

```
sudo yum install xorg*
```

4.5.3 Install GTK

First we need some libraries.

```
sudo yum install pango
sudo yum install pango-devel
sudo yum install cairo
sudo yum install glib2
sudo yum install redhat-lsb
sudo yum install libtiff
sudo yum install libtiff-devel
sudo yum install libjpeg-devel
sudo yum install gcc
```

Get ATK and ATK devel packages and install them.

```
cd /tmp
wget http://mirror.centos.org/centos/5/os/x86_64/CentOS/atk-1.12.2-1.fc6.x86_64.rpm
wget http://mirror.centos.org/centos/5/os/x86_64/CentOS/atk-devel-1.12.2-1.fc6.x86_64.rpm
sudo rpm -i atk-1.12.2-1.fc6.x86_64.rpm
sudo rpm -i atk-devel-1.12.2-1.fc6.x86_64.rpm
```

Get GTK2 and install it. The `make` and `make install` tasks will take a while, it took about 30 minutes. GTK2 will be stored in `/usr/local/lib`.

```
cd /tmp
wget http://ftp.gnome.org/pub/gnome/sources/gtk+/2.10/gtk+-2.10.14.tar.gz
tar -xvf gtk+-2.10.14.tar.gz
cd gtk+-2.10.14
sudo ./configure
sudo make
sudo make install
```

Change LD_LIBRARY_PATH so that **JOE** and **JID** can find the GTK. Edit `~/.bash_profile` for a persitent change.

```
...
if [ -n "$LD_LIBRARY_PATH" ]
then
    LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/usr/local/lib
else
    LD_LIBRARY_PATH=/usr/local/lib
fi

export LD_LIBRARY_PATH
```