Installation Guide

SafeArchive Version 2.0

Table of Contents

Conventions Used in This Guide

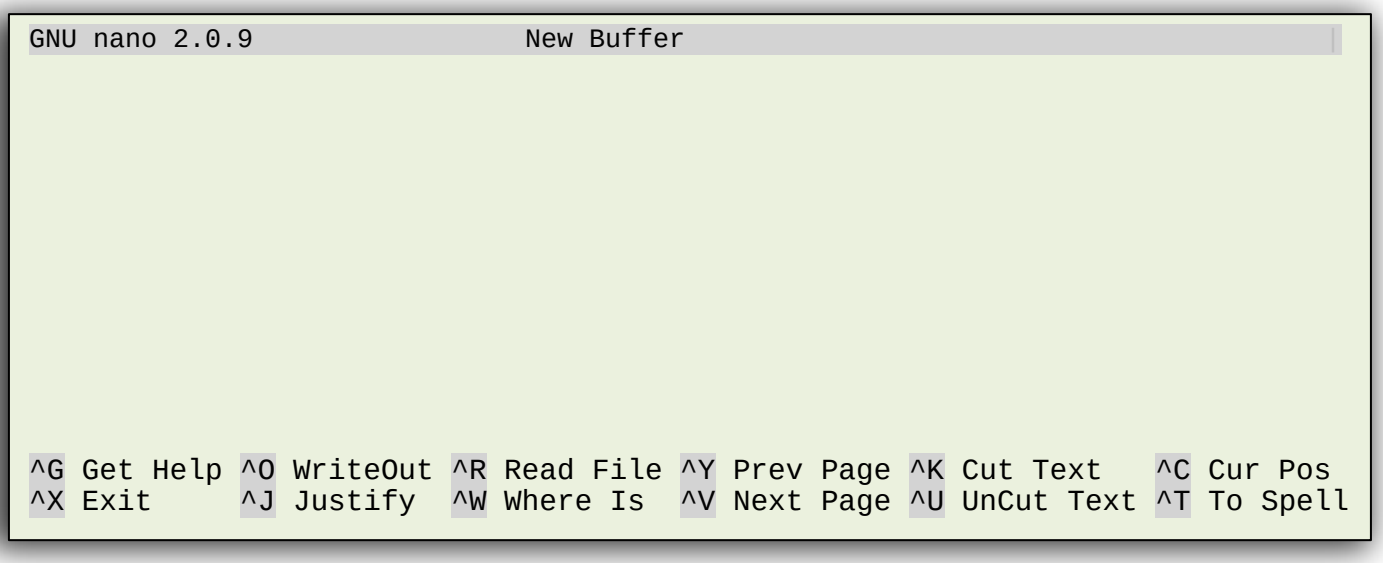
- Text, as in the figure below, appearing with a green highlight is meant to draw your attention to the highlighted word or phrase.

```
[root@localhost]# mysql –u root -p
```

- This guide assumes that the user has little to no experience with Unix or Unix-like operating systems.
- All text in `courier new` font, and in quotation marks like this: "`command`" is to be typed without the quotes.
- It is assumed that all text files will be edited with the Nano text editor. Don't let that stop you from using your favorite text editor.
- It is assumed that this installation will be performed by the root user. If you are not already the root user, you can elevate to root by typing the "`su –`" command at the command prompt, and entering your root password.

The Nano Text Editor

Most variants of Unix use either vi or Emacs. While vi and Emacs are powerful and extremely useful text editors, they lack the ease of use found in the Nano text editor.



Notes: To execute one of the functions listed on Nano's screen press "`ctrl + *`" where * is the substitution for letter next to the command. Another helpful hint is that WriteOut, simply means to save.

System Requirements

In order to use SafeArchive, you will need a private LOCKSS network. SafeArchive assumes that there is a private LOCKSS network (PLN) that can be configured to have it access the LOCKSS XML file via a web server. This configuration file is expected to have information about the LOCKSS-daemon debug account name and password shared by the PLN member servers.

SafeArchive also assumes that there is at least one actively running PLN member, a LOCKSS server that can be configured to have SafeArchive access its daemon-status data. Version 2.0 of SafeArchive does not have the capability to configure or start a PLN member server.

Hardware Requirements

Hardware	Standard Installation	Advanced Installation
CPU	Two cores @ 2.4 GHz	Four or more cores @ 2.4 GHz
Memory	4 GB	8 GB or more
Storage	60 GB	120 GB or more
Operating System	RHEL 6 / CEntOS 6	RHEL 6 / CEntOS 6

Table 1 -- Hardware Requirements

Software Dependencies

The table below lists the software required to perform a SafeArchive installation. Use the method outlined in **Table 2**, to obtain and install each of the software packages.

Software	Install Method	How to Obtain
MySQL 5.1.61	Yum	`yum install mysql` `yum install mysql-server`
Subversion	Yum	`yum install subversion`
Java JDK 6u37	Command Line	Download from: http://www.oracle.com/technetwork/java/javase/downloads/jdk6u37-downloads-1859587.html
MySQL Java Connector 5.1.*	Command Line	Download from: http://dev.mysql.com/downloads/connector/j
Apache Commons Logging 1.1.1	Command Line	Download from: http://commons.apache.org/logging/download_logging.cgi
Glassfish 3.1.2.2	Command Line	Download from: http://download.java.net/glassfish/3.1.2.2/release/glassfish-3.1.2.2-ml.zip

Table 2 – Software Dependencies

Software Installation

There are two ways to install SafeArchive. The first is to run the installer script found at safearchive.org. The second method is a lengthy procedure that allows more fine-tuning and control over your environment.

Script-based Installation

1. Download the install script from safearchive.org.
2. Unzip the saas2.zip file.
3. From the command line, and as root execute "`unzip saas2.zip`"
4. Type "`cd SAAS`" from the command line.
5. Run the prerequisites script with "`./prerequsites.sh`"
6. Run the installer script with "`./setup.sh`"
7. Answer the questions, and wait for the installer to finish.

Manual Installation

The software dependencies in **Table 2**, assume a base installation of RHEL 6 or CentOS 6. If you are unsure whether or not a software package is already installed, you can check by using the command, "`rpm –qa | grep <package-name>`". In the case of <package-name> it refers to the software for which you are searching.

If the software was not installed with RPM or Yum, you can use the `updatedb` and `locate` utilities to find out if these packages are installed. First run "`updatedb`" as root, to update the locate database. Then run "`locate <package-name> | grep bin`". This will show if any installed binaries exist for the package.

Create the gfish User

1. gfish, is the user that will run the Glassfish server. The root user should not be used to run Glassfish, and **will** cause problems if it is done. Additionally, all files in the glassfish application folder should be owned by gfish.

2. Create a user called gfish with the "`useradd`" command.

3. Set gfish's password with the "`passwd`" command.

```
[root@localhost]# useradd gfish
[root@localhost]# passwd gfish
New password:
Retype new password:
Passwd: all authentication tokens updated successfully.
```

Figure 1 -- You will not be able to see passwords as they are typed

Firewall Rules

It will be necessary to open a few TCP communications ports so that your server can be used and maintained. **Table 3**, below indicates the open ports that are required for SafeArchive to function properly.

TCP Port Number	Unix Application	Description
3306	MySQL	Database Server
3700	IIOP	Internet Inter-ORB Protocol
3820	IIOP/SSL	Secure version of Internet Inter-ORB Protocol
3920	IIOP/SSL with manual authentication	Secure version of Internet Inter-ORB Protocol with manual authentication
4848	Glassfish Administration Console	Tools used to configure Glassfish
8080	HTTP	Glassfish's Web server
8686	Pure JMX clients	Glassfish's JMX connector

Table 3 – Required open TCP port numbers for Glassfish

Begin by editing the /etc/sysconfig/iptables text file.

1. As root type "`cd /etc/sysconfig`" at the command prompt.
2. Type "`nano iptables`" at the command prompt.
3. Add the highlighted code in the figure below, paying attention to where it is placed.

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT –m state –state ESTABLISHED,RELATED –j ACCEPT
-A INPUT –p icmp –j ACCEPT
-A INPUT –i lo –j ACCEPT
-A INPUT –m state –-state NEW –m tcp –p tcp –-dport 22 –j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 3306 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 3700 -j ACCEPT
-A INPUT –m state –-state NEW –m tcp –p tcp –-dport 3820 –j ACCEPT
-A INPUT –m state –-state NEW –m tcp –p tcp –-dport 3920 –j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 4848 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 8080 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 8686 -j ACCEPT
-A INPUT –j REJECT –reject-with icmp-host-prohibited
-A FORWARD –j REJECT –reject-with icmp-host-prohibited
COMMIT
```

Figure 2 – Editing the firewall rules

4. The idea is to make sure everything added comes before the “REJECT” and “COMMIT” rules.

5. Once you have typed the code, save the text file by pressing “`ctrl + o`”.

6. Exit to the command prompt by pressing “`ctrl + x`”.

7. Restart the firewall with the `service` command. If you have done this correctly, you will see something similar to the figure below.

```
[root@localhost]# service iptables restart
iptables:  Flushing firewall rules:                    [  OK  ]
iptables:  Setting chains to policy ACCEPT: filter     [  OK  ]
iptables:  Unloading modules:                          [  OK  ]
iptables:  Applying firewall rules:                    [  OK  ]
```

Figure 3 – Restarting the iptables firewall with a valid iptables script

MySQL Database Client and Server

Yum can be used to install the MySQL client and server. Additionally, you will want make sure that the MySQL service will start when the server boots.

1. Install MySQL client with the command, “`yum install mysql`”

2. Install MySQL server with the command, “`yum install mysql-server`”

3. Begin the configuration of your MySQL server by issuing the following commands:

```
[root@localhost]# cp –p /etc/my.cnf /etc/my.cnf.orig
[root@localhost]# cp –fp /usr/share/mysql/my-medium.cnf /etc/my.cnf
```

Figure 4 – Installing the new MySQL configuration files, and backing up the default MySQL configuration files

4. Check that MySQL will start when the server boots, by issuing the following command: “`chkconfig | grep mysql`”

5. You should see output similar to the figure, below:

```
mysqld      0:off   1:off   2:off   3:off   4:off   5:off   6:off
```

Figure 5 – mysqld is not configured to start on boot

6. MySQL should start at runlevels 3 and 5. The figure above indicates that MySQL will not start at these runlevels. To correct this, issue a:
“`chkconfig mysqld on --level 35`” from the command line.

7. Confirm that MySQL will start at the correct runlevel by issuing the following command: "chkconfig | grep mysql"

8. You should see output similar to the figure, below.

```
mysqld      0:off   1:off   2:off   3:on    4:off   5:on    6:off
```

Figure 6 – mysqld is configured to start in text mode and graphical mode

9. The MySQL database server will now start when the server is booted. You can start it for the first time with the following command: "service mysqld start"

10. Next, we will create the SAAS database user, and secure the root password.

11. To test the MySQL connection, and create a MySQL root password; issue the following commands.

```
[root@localhost]# /usr/bin/mysqladmin –u root password $MySQLRootPW
[root@localhost]# mysql –u root -p
```

Figure 7 – Locking down MySQL on 127.0.0.1

12. You will be prompted to enter your root MySQL password. Once logged-in, you will see the MySQL prompt. Enter the commands in **Figure 8**, at the prompt.

Variable	Definition
$IP	The IP address of the Linux machine. If you are using an Amazon Web Instance, this will be your Elastic IP.
$MySQrootPW	Your MySQL root password
$SAAS_user	The username created for the SAAS database user.
$SAAS_pw	The password created for the SAAS database user

Table 4 -- MySQL user / password definitions

```
mysql> GRANT all ON *.* to 'root'@'$IP' IDENTIFIED BY '$MySQLrootPW';
Query OK, 0 rows affected (0.00 sec)
mysql> GRANT all ON safe_archive_system_db.* to '$SAAS_user'@'%' IDENTIFIED by '$SAAS_pw';
Query OK, 0 rows affected (0.00 sec)
mysql> GRANT all ON safe_archive_system_db.* to '$SAAS_user'@'127.0.0.1' IDENTIFIED by '$SAAS_pw';
Query OK, 0 rows affected (0.00 sec)
mysql> GRANT all ON safe_archive_system_db.* to '$SAAS_user'@'$IP' IDENTIFIED by '$SAAS_pw';
Query OK, 0 rows affected (0.00 sec)
mysql> GRANT all ON safe_archive_system_db.* to '$SAAS_user'@'localhost' IDENTIFIED by '$SAAS_pw';
Query OK, 0 rows affected (0.00 sec)
mysql> DELETE from mysql.user WHERE user='';
Query OK, 0 rows affected (0.00 sec)
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)
mysql> exit
```

Figure 8 – MySQL commands to create the MySQL root password, SAAS user, and SAAS user's password

Java JDK

You may obtain a copy of the Java JDK by directing your web browser to: http://www.oracle.com/technetwork/java/javase/downloads/jdk-6u26-download-400750.html.

To ensure compatibility with Glassfish 3.1.2.2, we are using Java version 6u37. Installation of the JDK will involve:

- Downloading the package
- Running the installer binary
- Creating the symbolic links to the Java binary and Java compiler
- Configuring the Java environment variables.

Note: If you already have a version of Java installed, these symbolic links may already exist.

1. On the Oracle website, select and download the binary that fits your system's platform. In this example we will use the x86/i586 based binary installer file.

2. In the folder that you downloaded the jdk-6u37-linux-*.bin file, issue the following commands:

```
[root@localhost]# cp jdk-6u37-linux-*.bin /opt
[root@localhost]# chmod 755 /opt/jdk-6u37-linux-*.bin
[root@localhost]# ./jdk-6u37-linux-i586.bin
```

Figure 9 – Installing the JDK

3. Wait until all files are extracted. Depending on the speed of your system, this could take a while.

4. Create the symbolic links. If the symbolic links already exist for java and javac, you will need to rename them first with the mv command.

```
[root@localhost]# mv /usr/bin/java /usr/bin/java.bak
[root@localhost]# mv /usr/bin/javac /usr/bin/javac.bak
[root@localhost]# ln -s /opt/jdk1.6.0_37/bin/java /usr/bin/java
[root@localhost]# ln -s /opt/jdk1.6.0_37/bin/javac /usr/bin/javac
```

Figure 10 – Creating the symbolic links to the java binary and the java compiler

Note: An alternative to setting the JAVA_HOME environment variable is to include the path to Java's application folder in the Glassfish configuration file, asconfig.conf. Add the variable, AS_JAVA="/opt/jdk1.6.0_37" to the end of the configuration file. It is recommended that JAVA_HOME be set in the .bash_profile, so that java will work correctly with applications other than Glassfish.

5. Add the JAVA_HOME environment variables to the .bash_profile in root's home folder and gfish's home folder. Make sure to edit gfish's .bash_profile as gfish (see Figure 11).

```
[root@localhost]# su gfish -c "nano ~/.bash_profile"
```

Figure 11 – Starting the Nano text editor as the gfish user

6. The ~/bash_profile files should look something like this when complete. The green highlighting indicates where to add the JAVA_HOME environment variables and path changes.

```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs

JAVA_HOME="/opt/jdk1.6.0_37"
export JAVA_HOME

PATH=$PATH:$HOME/bin:$JAVA_HOME:$JAVA_HOME/bin

export PATH
```

Figure 12 – An example of the .bash_profile once the JAVA_HOME environment variables have been added

7. Once .bash_profile has been changed and saved, exit the text editor and execute "`source ~/.bash_profile`" as root. This will activate the new environment variables.

Glassfish Installation

The Installation of Glassfish Application Server will involve:

- Downloading the package
- Unpacking the software
- Editing the XML configuration files
- Installing the init script
- Configuring the application server for SafeArchive

Like the Java JDK, Glassfish Application Server can be downloaded from Oracle. The URL is: `http://download.java.net/glassfish/3.1.2.2/release/glassfish-3.1.2.2-ml.zip`

1. Download the Glassfish Application Server from the URL above.

2. Once you have downloaded the file, issue the following commands from your download location, to install the Glassfish Application Server:

```
[root@localhost]# cp glassfish-3.1.2.2-ml.zip /usr/local
[root@localhost]# cd /usr/local
[root@localhost]# unzip glassfish-3.1.2.2-ml.zip
[root@localhost]# rm glassfish-3.1.2.2-ml.zip
```

Figure 13 – Installing Glassfish Application Server

3. Your Glassfish Application Server is installed.

SafeArchive XML Configuration

1. Sourceforge, can be used to obtain the SafeArchive XML configuration files. The URL is: http://safearchive.svn.sourceforge.net/viewvc/safearchive/trunk/configurationFiles/

2. Download saas-audit-config.xml and saas-local-config.xml to: `/usr/local/glassfish3/glassfish/domains/domain1/config`

3. Begin by issuing the command: "`nano saas-local-config.xml`" as the root user, from the download directory.

4. The figure below is an excerpt from `saas-local-config.xml`. Edit your file so that the green highlighted text reflects your LOCKSS environment. Save the file with "`ctrl + x`".

```
...
<entry key="saas.support.contactus.emailaddress">joe@domain.edu</entry>
...
<entry key="saas.lockss.xml.url">host.domain.edu/lockss.xml</entry>
...
<entry key="saas.daemonstatus.account">debug</entry>
...
<entry key="saas.support.contactus.emailrecipientname">Joe User</entry>
<entry key="saas.captcha.privatekey">SABAADx7qwLkzL</entry>
...
<entry key="saas.daemonstatus.password">password</entry>
```

Figure 14 -- Excerpt from saas-local-config.xml

Key	Definition
saas.support.contactus.emailaddress	The email address of the person supporting your SafeArchive system.
saas.lockss.xml.url	The full path to locks.xml.
saas.daemonstatus.account	Your PLN's common debug account.
saas.support.contactus.emailrecipientname	The name of the person supporting your SafeArchive system.
saas.captcha.privatekey	Google reCAPTCHA private key.
saas.daemonstatus.password	The password for the common debug account.

Table 5 – Key definitions for saas-local-config.xml

5. Issue: "`nano saas-local-config.xml`" as the root user, from the download directory.

6. The figure below is an excerpt from `saas-audit-config.xml`. Edit your file so that the green highlighted text reflects your LOCKSS environment. Save the file with "`ctrl + x`".

```
...
<entry key="saas.audit.report.emailnotice.recipients">joe@domain.edu,
mike@domain.edu</entry>
...
<entry key="saas.audit.report.birtviewer.servername">SAAS.domain.edu</entry>
...
```

Figure 15 -- – Excerpt from saas-audit-config.xml

Key	Definition
saas.audit.report.emailnotice.recipients	Your audit-report recipients' email addresses (comma-separated list).
saas.audit.report.birtviewer.servername	The fully qualified domain name of the SafeArchive system, or its IP address.

Figure 16 --Key definitions for saas-audit-config.xml

Glassfish Init Installation

You will need to make sure that the Glassfish server will start at boot. No init script (start-up script) is included with this distribution of Glassfish. Use the one in the appendix, or alternatively write your own. http://www.cyberciti.biz/tips/linux-write-sys-v-init-script-to-start-stop-service.html has a great tutorial for writing chkconfig compatible, System V style startup scripts.

1. If you use the init script from the appendix, copy it to /etc/init.d, change the permissions, and install the script with the chkconfig command (see the figure below).

```
[root@localhost]# cp glassfish /etc/init.d
[root@localhost]# chmod 755 /etc/init.d/glassfish
[root@localhost]# chkconfig glassfish on --level 35
```

Figure 17 -- Installing the Glassfish init script

2. Confirm that Glassfish will start at the correct runlevel by issuing the following command: "`chkconfig | grep glassfish`"

3. You should see output similar to the figure, below.

```
glassfish  0:off   1:off   2:off   3:on   4:off   5:on   6:off
```

Figure 18 -- glassfish is configured to start in text mode and graphical mode

Additional software is required to help Glassfish communicate with the MySQL server (MySQL Java Connector), and to provide a log interface for SAFE Archive Audit System (Apache Commons Logging).

MySQL Java Connector

1. Download the MySQL Java Connector from: http://dev.mysql.com/downloads/connector/j

2. Run the following commands from your download folder, as the root user:

```
[root@localhost]# unzip mysql-connector-java*.zip
[root@localhost]# cd mysql-connector-java-5.1.21
[root@localhost]# export gf_dir="/usr/local/glassfish3/glassfish"
[root@localhost]# cp mysql-connector-java-*.jar $gf_dir/domains/domain1/lib
```

Figure 19 -- Installing the MySQL Java Connector

Apache Commons Logging

1. Download the Apache Commons Logging jar file from:
   http://commons.apache.org/logging/download_logging.cgi

2. Run the following commands from your download folder, as the root user:

```
[root@localhost]# unzip commons-logging-1.1.1.zip
[root@localhost]# cd commons-logging-1.1.1
[root@localhost]# cp commons-logging-*.jar $gf_dir/domains/domain1/lib
```

Figure 20 – Installing the Apache Commons Logging

3. The last step is to change the ownership of all files in the Glassfish installation directory.

```
[root@localhost]# chown –R gfish /usr/local/glassfish3
[root@localhost]# chgrp –R gfish /usr/local/glassfish3
```

Figure 21 -- Changing the ownership on the Glassfish install directory

Before attempting to start the Glassfish server, you will want to ensure that the /etc/hosts file is configured properly; otherwise you may have issues starting the Glassfish domain. For this example, let's assume our Red Hat Linux Enterprise's machine name is SAAS, its domain name is domain.edu, and its IP address is 10.10.10.10.

1. Begin by issuing the command: "`nano /etc/hosts`" as the root user.

2. Examine this file, and make sure there is an entry for your Linux box (IP, hostname, FQDN). See below.

```
127.0.0.1       localhost  localhost.localdomain
10.10.10.10      SAAS     SAAS.domain.edu
```

Figure 22 -- Example of a properly configured /etc/hosts file

Note: FQDN = Fully Qualified Domain Name (ex. SAAS.domain.edu).

Setting the Admin Password

1. You are now ready to start the Glassfish server. Refer to the figure below.

```
[root@SAAS.domain.edu]# service glassfish start
Starting Glassfish: Waiting for domain1 to start .............
Successfully started the domain : domain1
domain  Location: /usr/local/glassfish3/glassfish/domains/domain1
Log File: /usr/local/glassfish3/glassfish/domains/domain1/logs/server.log
Admin Port: 4848
Command start-domain executed successfully.
```

Figure 23 – Starting the Glassfish server with the service command

2. Change the admin password with the following command. This example uses the password: "adminadmin" as the new password.

```
[root@SAAS.domain.edu]# cd /usr/local/glassfish3/glassfish/bin
[root@SAAS.domain.edu]# su gfish -c "./asadmin"
asadmin> change-admin-password
Enter admin user name [default: admin]> admin
Enter admin password>
Enter new admin password>
Enter new admin password again>
Command change-admin-password executed successfully.
asadmin> enable-secure-admin
Enter admin user name> admin
Enter admin password for user "admin">
You must restart all running servers for the change in secure admin to take
effect.
Command enable-secure-admin executed successfully.
asadmin> restart-domain
Successfully restarted the domain
Command restart-domain executed successfully.
```

Figure 24 – Changing the admin password (passwords are not visible as they are typed)

3. Open a web browser and point it to tcp port 4848 of your Glassfish server.

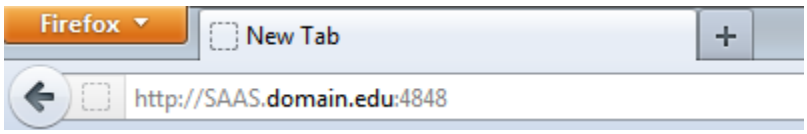


Figure 25 -- Browsing to the admin console for Glassfish with Firefox

4. After the login page loads, login to the Glassfish admin console with the following credentials.

   User Name: admin
   Password: adminadmin

Figure 26 -- Logging into Glassfish

JVM Configuration

1. On the left side of the screen, find the navigation pane. Browse to: **Configurations** ➡ **default-config** ➡ **JVM Settings**. See **Figure 27**, below.

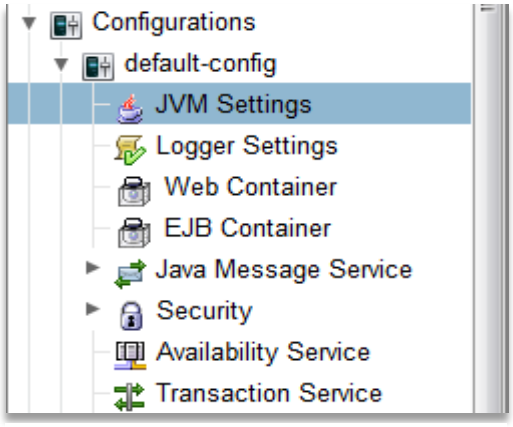


Figure 27 -- Finding the JVM Settings

2. Under JVM Settings, you should see a tab labeled **JVM Options**. Click this tab.

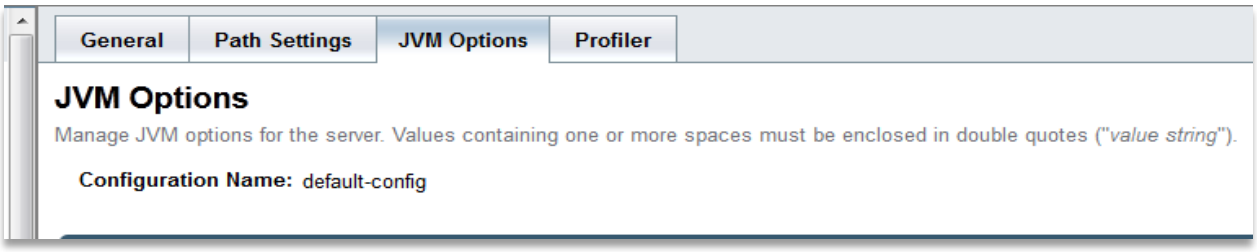


Figure 28 -- Finding the JVM Options

3. Click the **Add JVM Option** button six times. Six new text boxes will appear in the default-config.

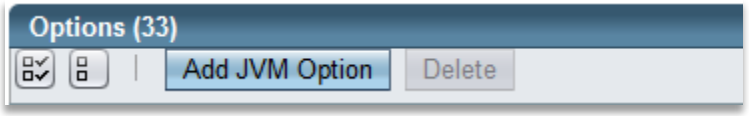


Figure 29 -- Add JVM Option

4. Add the following parameters to the new text boxes in the options list. Reference the figure below, if you need an example of the values for the **LOCKSS XML URL** or the **host captcha private key**. These two values will depend on your individual settings.

JVM Options
-Dsaas.audit.config.file=**${com.sun.aas.instanceRoot}/config/saas-audit-config.xml**
-Dsaas.captcha.privatekey=***Your host captcha private key*** **(**Google reCAPTCHA key, see Appendix B**)**
-Dsaas.ejb.jndi.modulename=**java:global/safearchiveauditsystem-ear/safearchiveauditsystem-ejb-1.1/**
-Dsaas.jdbc.jndi.name=**jdbc/saasDS**
-Dsaas.localconfig.file=**${com.sun.aas.instanceRoot}/config/saas-local-config.xml**
-Dsaas.lockss.xml.url=***Your LOCKSS XML URL***

Table 6 -- JVM Options

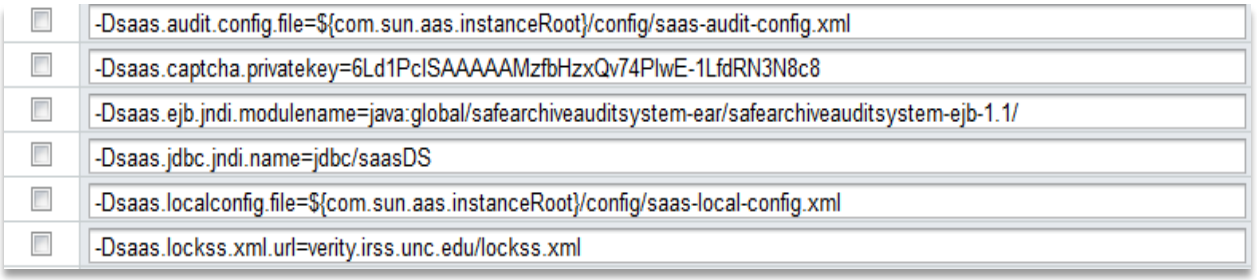


Figure 30 -- Configuring the JVM Options

5. Once you have entered all the values for the JVM Options, click the **Save** button in the upper-right of the screen. If the save is successful, you will see the message displayed in **Figure 32**.

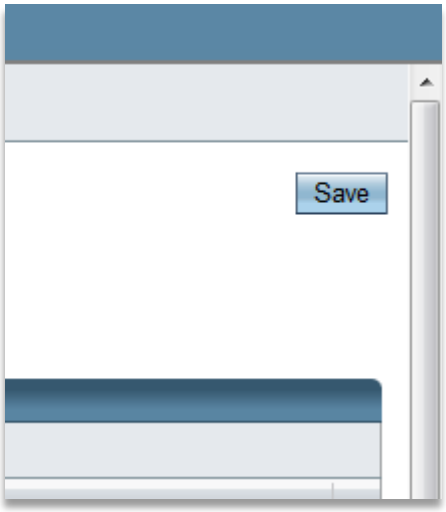


Figure 31 – Saving the JVM Options

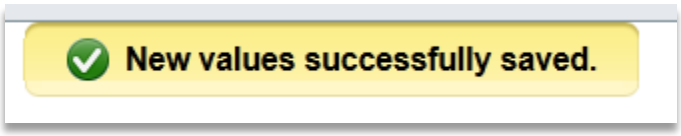


Figure 32 – JVM Options have been saved

JDBC Connection Pool Configuration

1. On the left side of the screen, find the navigation pane. Browse to: **Resources** ➡ **JDBC** ➡ **JDBC Connection Pools**. See Figure 33, below.

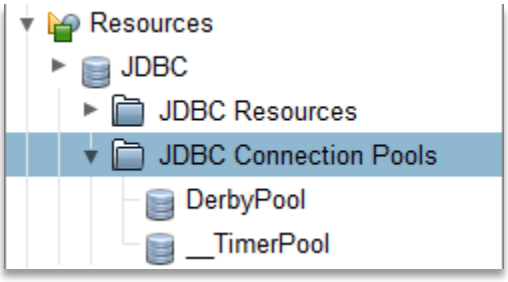


Figure 33 -- Finding the JCBC Connection Pool settings

2. On the right side of the screen, click the **New** button.

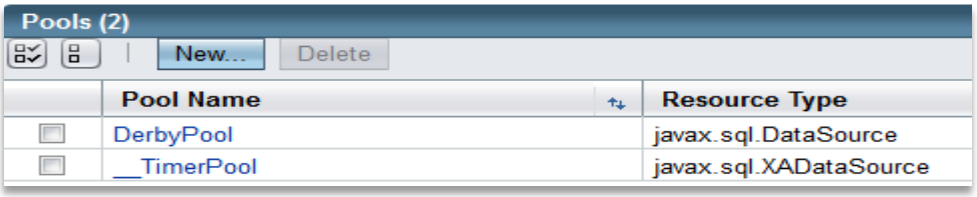


Figure 34 -- New JDBC Connection Pool

3. Enter the following parameters and click **Next**:
    - **Pool Name**: saasDbPool
    - **Resource Type**: java.sql.DataSource
    - **Database Driver Vendor**: MySql

Figure 35 -- JDBC Connections Pool step 1 of 2

4. Scroll to the bottom of the page, and add the following values to Additional Properties. You should only need to click the **Add Property** button twice for Url and URL. All other fields should exist. Consult **Figure 36**, if you need an example of the input value.

Parameter	Value
Url	jdbc:mysql://***Your DB Server's IP Address***:3306/safe_archive_system_db
URL	jdbc:mysql:// ***Your DB Server's IP Address***:3306/safe_archive_system_db
databaseName	safe_archive_system_db
serverName	Your DB Server's IP Address
user	The SafeArchive user created in the MySQL section
Password	The password for the SAAS user created in the MySQL section

Table 7 -- JDBC parameters and values

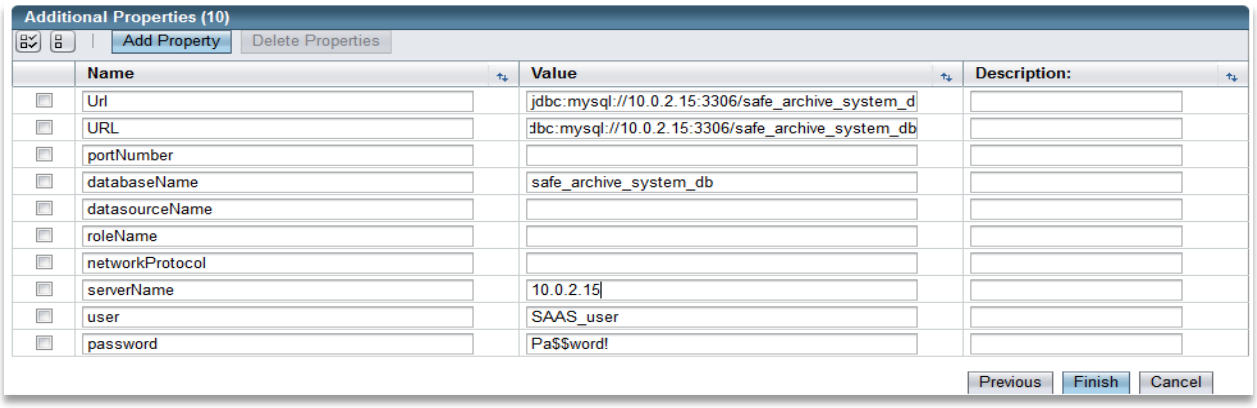


Figure 36 -- JDBC Connections Pool step 2 of 2

5. Click **Finish** to complete the JDBC Connection Pool configuration.

JDBC Resource Configuration

1. On the left side of the screen, find the navigation pane. Browse to: **Resources** ➡ **JDBC** ➡ **JDBC Resources**.

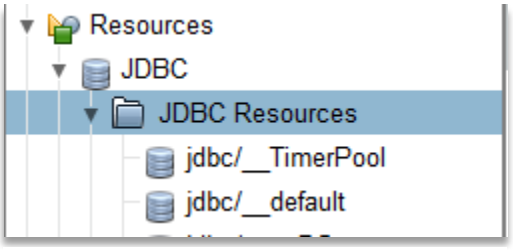


Figure 37 – Finding JDBC Resources

2. On the right side of the screen, click the **New** button.

Figure 38 – New JDBC Resource

3. On the right of the screen, type the parameters listed below.

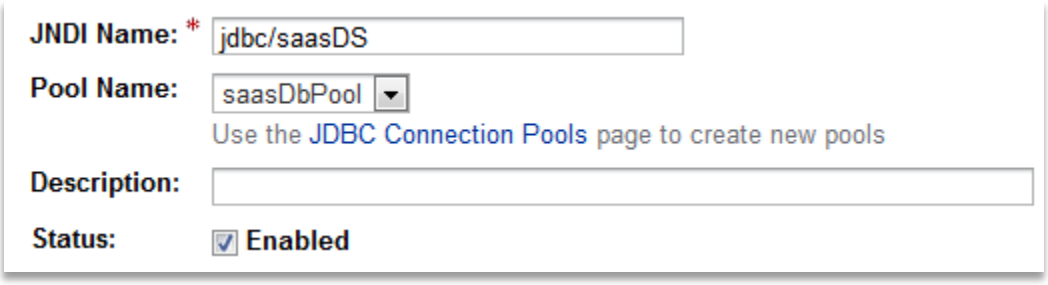


Figure 39 – New JDBC Resource

4. Click **OK** to save.

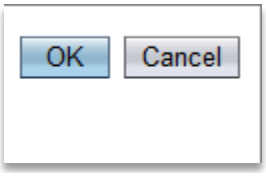


Figure 40 – Saving New JDBC Resource

JavaMail Session Configuration

1. On the left side of the screen, find the navigation pane. Browse to: **Resources** ➡ **JavaMail Sessions**. See below.

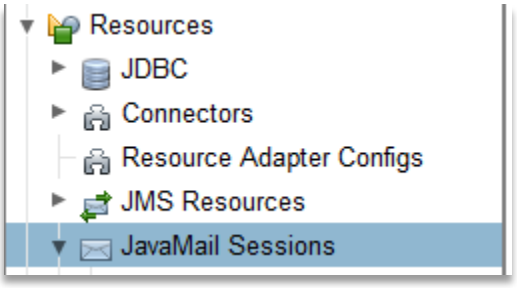


Figure 41 -- Finding JavaMail Sessions

2. On the right side of the screen, click the **New** button.

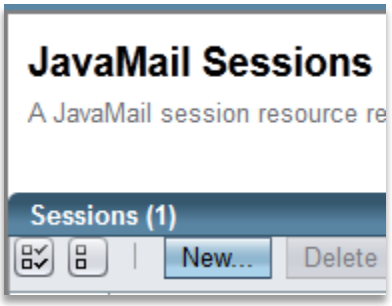


Figure 42-- New JavaMail Session

3. On the right of the screen, type in the parameters listed below.

Parameter	Definition
JNDI Name	A meaningful name for the JavaMail session. Use the default in the figure on the next page (mail/saasMailSession).
Mail Host	This is the IP address or host name of your mail server that sends mail.
Default User	The username of SMTP user. This is usually just the first part of your email address, before the @ sign.
Default Sender Address	This is the email address associated with the SMTP account.
Description	A helpful description for the JavaMail session.

Table 8 – JavaMail Session parameter definitions

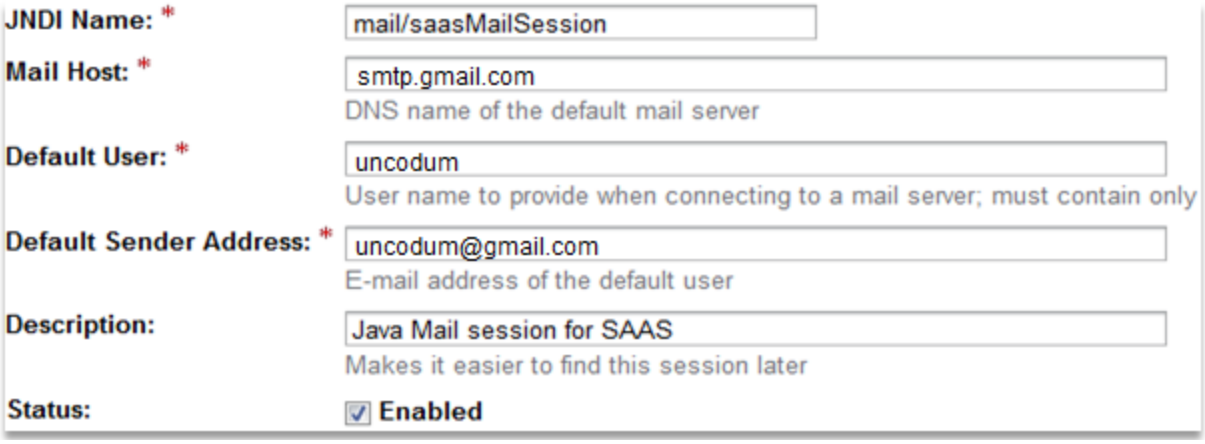


Figure 43 -- JavaMail Parameters

4. Scroll down to the bottom of the screen, and add the six properties listed below.

Key	Definition
mail.smtp.socketFactory.port	TCP port running SMTP on your mail server. Usually 25, 465 for Gmail.
mail.smtp.port	TCP port running SMTP on your mail server.
mail.smtp.socketFactory.fallback	Set to **false**
mail.smtp.auth	Set to **true**
mail.smtp.password	The password associated with the SMTP user.
mail.smtp.socketFactory.class	Set to **javax.net.ssl.SSLSocketFactory**

Table 9 – Java Mail Settings

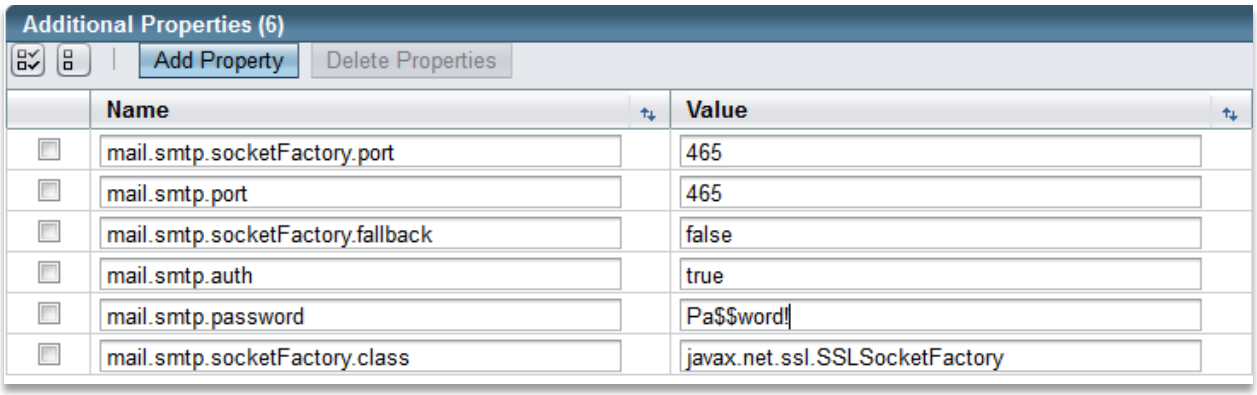


Figure 44 -- JavaMail Session Properties

5. Click **OK** to save.

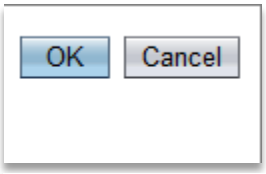


Figure 45 – Saving New JavaMail Session

JMS Resource Configuration

The SAFE Audit Archive System (SAAS), version 2.0 uses two JMS connection factories and two destination resources.

1. Begin the configuration, by browsing to **Resources** ➡ **JMS Resources** ➡ **Connection Factories**

2. On the right side of the screen, click the **New** button under JMS Connection Factories.

Figure 46 – New JMS Connection Factories

3. The New JMS Connection Factory page will open.

4. Type the following data in the General Settings pane:

Pool name: jms/SAASPostAuditReportActionQueueConnectionFactory
Resource type: javax.jms.QueueConnectionFactory

5. Click the **OK** button.

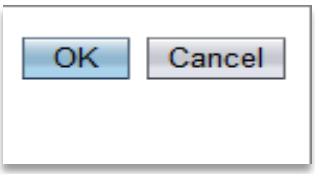


Figure 47 – Saving New JMS Connection Factory

6. Repeat the above steps 2 - 5 with the following data:

Pool name: jms/SAASPostETLActionQueueConnectionFactory
Resource type: javax.jms.QueueConnectionFactory

7. Browse to **Resources** ➡ **JMS Resources** ➡ **Destination Resources**

8. On the right side of the screen, click the **New** button under JMS Destination Resources.

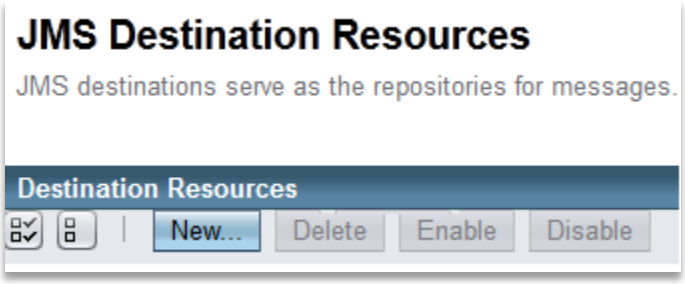


Figure 48 – New JMS Destination Resources

9. The New JMS Destination Resource page will open.

10. Type the following data for each respective field:

JNDI Name: jms/SAASPostAuditReportAction
Physical Destination Name: SAASPostAuditReportAction
Resource Type: javax.jms.Queue

11. Click the **OK** button.

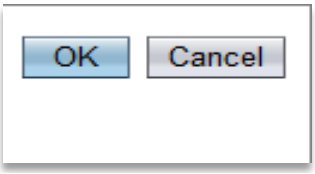


Figure 49 – Saving New JMS Connection Factory

12. Repeat the above steps 7 - 11 with the following data:

JNDI Name: jms/SAASPostETLAction
Physical Destination Name: SAASPostETLAction
Resource Type: javax.jms.Queue

User Authentication Configuration

The SAFE Audit Archive System (SAAS) uses the file-realm of the Glassfish Application Server as its authentication mechanism. SAAS pre-defines two security roles: **Administrator** and **Curator**. SAAS also pre-defines two corresponding security groups: **saasAdminGroup** and **saasCuratorGroup**.

To log into SAAS, a new user must be registered in Glassfish, and each user must be mapped to an appropriate role. You will need to create the three users, listed in **Table 10** on page 24.

1. Begin the configuration, by browsing to **Configurations** ➡**server-config** ➡**Security** ➡ **Realms** ➡ **file**

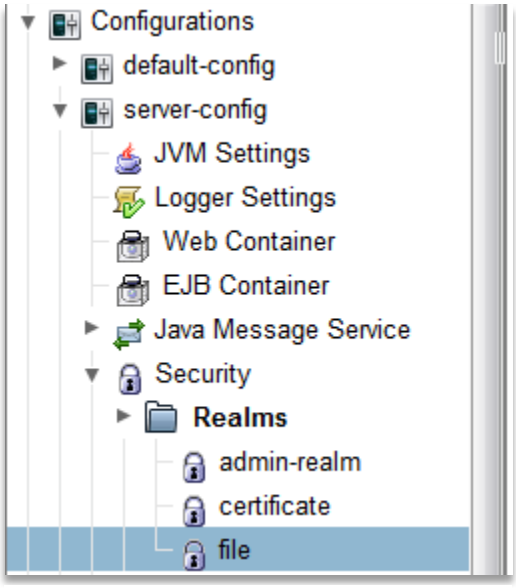


Figure 50

2. On the right side of the screen, click the **Manage Users** button.

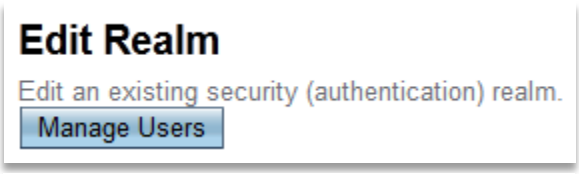


Figure 51 -- Editing the file Security Realm

3. Click the **New** button.

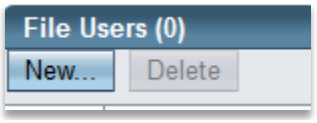


Figure 52 -- Creating a new user

4. Fill in the fields as seen in the figure below and click **OK**. Repeat the process to create additional users.

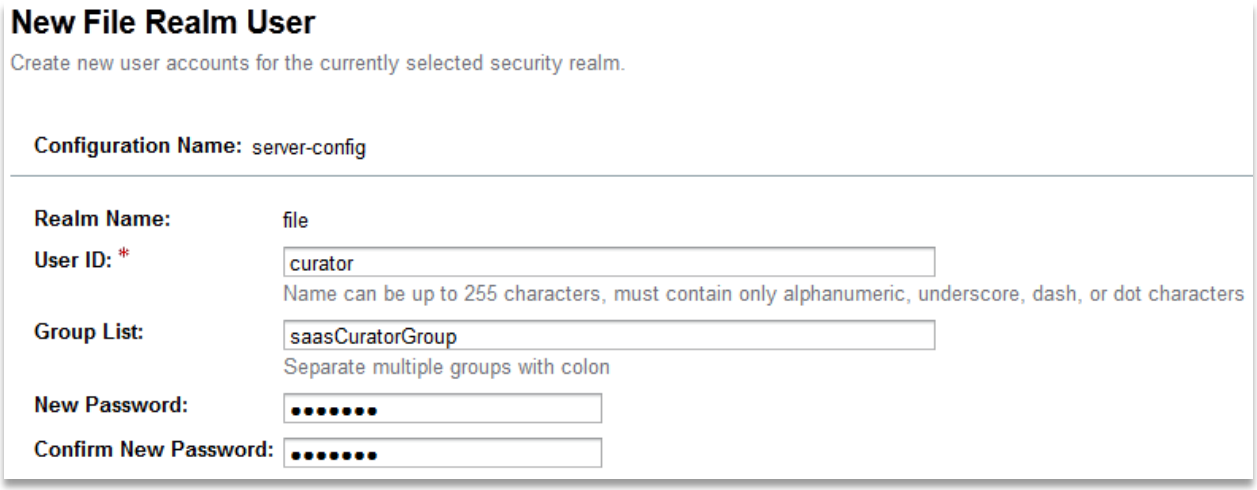


Figure 53 -- Creating a new user

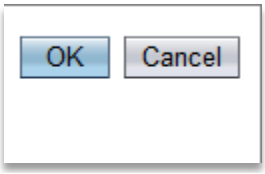


Figure 54 – Saving New Users

SafeArchive Username	SafeArchive Group	SafeArchive Password
curator	saasCuratorGroup	curator
birtviewer	saasCuratorGroup	birtviewer
administrator	saasAdminGroup	administrator

Table 10 – SafeArchive file realm, user table

Subversion

Begin the subversion installation and configuration with yum.

1. If you are not already the root user, you can elevate to root by typing the "`su -`" command at the command prompt, and entering your root password.

2. Install subversion with the following command "`yum install subversion`"

3. Subversion is now installed. Verify that Subversion is installed by typing the following at the command prompt: "`svn --version`"

4. Create the following directories:

    a. /usr/safe/data/audit_input_files
    b. /usr/safe/data/audit_output_files

```
[root@SAAS.domain.edu]# cd /usr
[root@SAAS.domain.edu]# mkdir safe
[root@SAAS.domain.edu]# cd safe
[root@SAAS.domain.edu]# mkdir data
[root@SAAS.domain.edu]# cd data
[root@SAAS.domain.edu]# mkdir audit_input_files
[root@SAAS.domain.edu]# mkdir audit_output_files
```

Figure 54 – Creating input and output directories for subversion

5. Direct your browser to Sourceforge to:
   http://safearchive.svn.sourceforge.net/viewvc/safearchive/trunk/configurationFiles/

6. Download the following files:

   - SSPSchema_v1_05.xsd
   - locks_audit_diff_xpaths.xml
   - locks_audit_report.xsd
   - us-state-code-to-label.properties
   - us-state-code-to-region-code.properties
   - us-census-region-modified-code-to-label.properties
   - saas-audit-subject-set.txt
   - saas-audit-owner-inst-set.txt

7. Copy all the files into **/usr/safe/data/audit_input_files**:

```
[root@SAAS.domain.edu]# cp *.xsd /usr/safe/data/audit_input_files
[root@SAAS.domain.edu]# cp saas*.txt /usr/safe/data/audit_input_files
[root@SAAS.domain.edu]# cp *.xml /usr/safe/data/audit_input_files
[root@SAAS.domain.edu]# cp *.properties /usr/safe/data/audit_input_files
```

Figure 556 – Copying files to /usr/safe/data/audit_input_files

8. Create a Subversion repository on the server with the commands:

```
[root@SAAS.domain.edu]# svnadmin create /usr/safe/svn
[root@SAAS.domain.edu]# svn import /usr/safe/data \
> file:///usr/safe/svn/trunk -m "initial import"
Adding         /usr/safe/data/audit_input_files
Adding         /usr/safe/data/audit_input_files/saas-audit-subject-set.txt
Adding         /usr/safe/data/audit_input_files/us-state-code-to-region-code.propert
Adding         /usr/safe/data/audit_input_files/saas-local-config.xml
Adding         /usr/safe/data/audit_input_files/lockss_audit_diff_xpaths.xml
Adding         /usr/safe/data/audit_input_files/us-census-region-modified-code-to-
Adding         /usr/safe/data/audit_input_files/us-state-code-to-label.properties
Adding         /usr/safe/data/audit_input_files/saas-audit-owner-inst-set.txt
Adding         /usr/safe/data/audit_input_files/lockss_audit_report.xsd
Adding         /usr/safe/data/audit_input_files/saas-audit-config.xml
Adding         /usr/safe/data/audit_input_files/SSPSchema_v1_05.xsd
Adding         /usr/safe/data/audit_output_files
```

Figure 567 – Creating a repository for subversion

SafeArchive Installation

Now we can install the application files for SafeArchive.

1. Download the following files from Sourceforge:

File	Location
birt.war	http://safearchive.svn.sourceforge.net/viewvc/safearchive/trunk/configurationFiles/
safearchiveauditsystem-ear.ear	http://safearchive.svn.sourceforge.net/viewvc/safearchive/trunk/ear/
trac_qq.txt	http://safearchive.svn.sourceforge.net/viewvc/safearchive/trunk/configurationFiles/

Table 11 -- SafeArchive install files and download locations

2. In the Glassfish Admin console, click on **Applications** in the navigation tree:

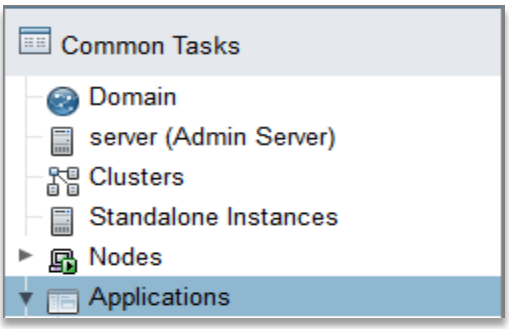


Figure 578-- Finding the Applications node in Glassfish

3. Click on the **Deploy** button, on the right.

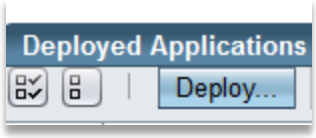


Figure 589-- Deploying an application in Glassfish

4. Click **Browse**, and find the downloaded birt.war file.

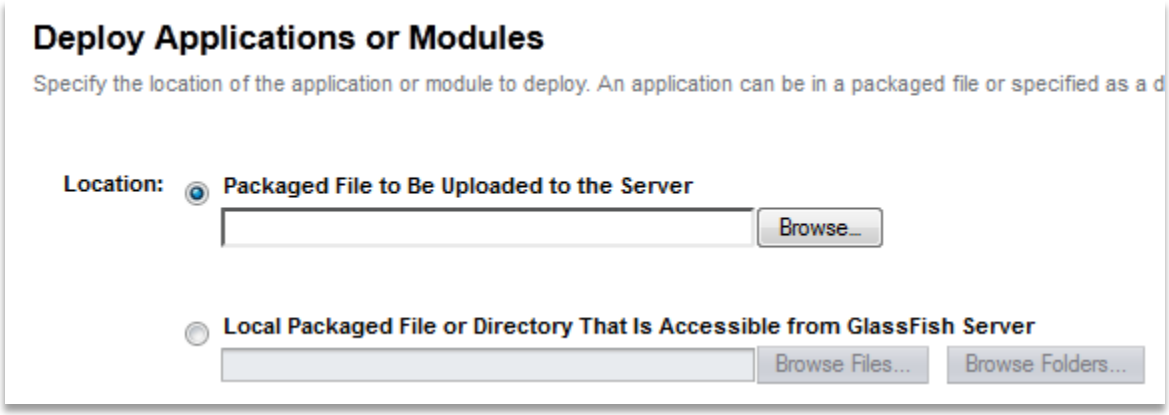


Figure 60-- Browsing for applications in Glassfish

5. Next, Click **OK**.

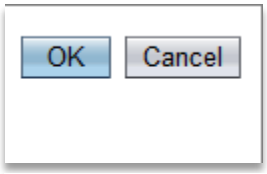


Figure 59 – Deploying the .war and .ear files

6. Repeat the process in steps 3 – 5 for the .ear file.
7. Once both files are deployed, browse to the directory where trac_qq.txt is located.
8. Login to MySQL, with the MySQL root password and issue the following commands.

   Note: This is the same password set in the MySQL section.

```
[root@SAAS.domain.edu]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 922
Server version: 5.1.52-log Source distribution

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights
reserved.
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL v2
license

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

mysql> LOAD DATA LOCAL INFILE "trac_qq.txt" INTO TABLE
-> trac_audit_checklist_criteria_data (aspect_id, criterion);
```

Figure 62 – Post install MySQL configuration

9. Your SafeArchive system is now installed. Direct your web browser to:

   http://$IP:8080/safearchiveauditsystem-web/faces/index.xhtml

   Note: $IP is the FQDN or IP address of your SafeArchive system.

Appendix A. – Glassfish init Script

```
#!/bin/bash

# chkconfig: - 99 14
# description: Glassfish Application Server
# processname: glassfish


#############################################################
# Glassfish 3.1.2.2 startup script                          #
#                                                           #
# By: Terry Rowland - trowland@email.unc.edu                #
#                                                           #
# Based on freshclam init by:                               #
# (c) 2004/05/17 Petr@Kristof.CZ under GNU GPL 2.0+         #
#                                                           #
# Updated 10/25/2012 to accommodate Glassfish 3.1.2.2       #
#                                                           #
#############################################################

# Source function library
. /etc/init.d/functions

# Get network config
. /etc/sysconfig/network

RETVAL=0

start() {
    echo -n $"Starting Glassfish: "
    # Start me up!
    su gfish -c '/usr/local/glassfish3/bin/asadmin start-domain'
    RETVAL=$?
    echo
    [ $RETVAL -eq 0 ] && touch /var/lock/subsys/glassfish
    return $RETVAL
}

stop() {
    echo -n $"Stopping Glassfish: "
    su gfish -c '/usr/local/glassfish3/bin/asadmin stop-domain'
    RETVAL=$?
    echo
    [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/glassfish
    return $RETVAL
}

status() {
    echo -n $"Getting Glassfish Status... "
    echo ""
    su gfish -c '/usr/local/glassfish3/bin/asadmin list-domains'
    RETVAL=$?
    echo
    return $RETVAL

}

restart() {
    stop
    start
}
```

```
reload() {
    stop
    start
}

case "$1" in
  start)
      start
    ;;
  stop)
      stop
    ;;
  status)
      status
    ;;
  restart)
      restart
    ;;
  condrestart)
      [ -f /var/lock/subsys/glassfish ] && restart || :
    ;;
  reload)
    reload
    ;;
  *)
    echo $"Usage: $0 {start|stop|restart|condrestart|reload}"
    exit 1
esac

exit $?
```

Appendix B – Google reCAPTCHA

CAPTCHA stands for "**C**ompletely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part." SafeArchive uses this technology, in the form of Google reCAPTCHA private keys. This helps to keep spammers from abusing the email forms in the SafeArchive support feature.

You will need a Google account before using Google reCAPTCHA. If you do not already have a Google account, go to google.com, and follow the directions below.

1. Go to https://www.google.com/recaptcha
2. Click on **USE reCAPTCHA ON YOUR SITE**.

Figure 60 – USE reCAPTCHA ON YOUR SITE

3. Next click **Sign up Now!**

Figure 61 – Sign Up Now

4. Type in the domain of your website, and click **Create Key**.

Figure 62 -- Setting CAPTCHA domain

5. Use the newly generated private key in the JVM Configuration.

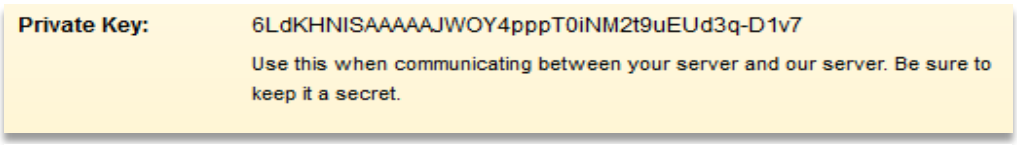


Figure 63-- New CAPTCHA private key

Index

S

T

U

V

Y