Red Hat Enterprise MRG 2
Management Console Installation Guide

Installing and Configuring the MRG Management Console

David Ryan

Red Hat Enterprise MRG 2 Management Console Installation Guide

Installing and Configuring the MRG Management Console

David Ryan
Red Hat Engineering Content Services
dryan@redhat.com

Legal Notice

Keywords

Abstract

This guide covers the installation and configuration of the MRG Management Console.

Table of Contents

Preface

1. Document Conventions

This manual uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

In PDF and paper editions, this manual uses typefaces drawn from the Liberation Fonts set. The Liberation Fonts set is also used in HTML editions if the set is installed on your system. If not, alternative but equivalent typefaces are displayed. Note: Red Hat Enterprise Linux 5 and later includes the Liberation Fonts set by default.

1.1. Typographic Conventions

Four typographic conventions are used to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

`Mono-spaced Bold`

Used to highlight system input, including shell commands, file names and paths. Also used to highlight keys and key combinations. For example:

> To see the contents of the file **`my_next_bestselling_novel`** in your current working directory, enter the **`cat my_next_bestselling_novel`** command at the shell prompt and press **`Enter`** to execute the command.

The above includes a file name, a shell command and a key, all presented in mono-spaced bold and all distinguishable thanks to context.

Key combinations can be distinguished from an individual key by the plus sign that connects each part of a key combination. For example:

> Press **`Enter`** to execute the command.
>
> Press **`Ctrl`**+**`Alt`**+**`F2`** to switch to a virtual terminal.

The first example highlights a particular key to press. The second example highlights a key combination: a set of three keys pressed simultaneously.

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in **`mono-spaced bold`**. For example:

> File-related classes include **`filesystem`** for file systems, **`file`** for files, and **`dir`** for directories. Each class has its own associated set of permissions.

Proportional Bold

This denotes words or phrases encountered on a system, including application names; dialog box text; labeled buttons; check-box and radio button labels; menu titles and sub-menu titles. For example:

> Choose **System** → **Preferences** → **Mouse** from the main menu bar to launch **Mouse Preferences**. In the **Buttons** tab, click the **Left-handed mouse** check box and click **Close** to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).
>
> To insert a special character into a **gedit** file, choose **Applications** → **Accessories** →

> **Character Map** from the main menu bar. Next, choose **Search** → **Find…** from the **Character Map** menu bar, type the name of the character in the **Search** field and click **Next**. The character you sought will be highlighted in the **Character Table**. Double-click this highlighted character to place it in the **Text to copy** field and then click the **Copy** button. Now switch back to your document and choose **Edit** → **Paste** from the **gedit** menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in proportional bold and all distinguishable by context.

Mono-spaced Bold Italic or ***Proportional Bold Italic***

Whether mono-spaced bold or proportional bold, the addition of italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

> To connect to a remote machine using ssh, type **ssh *username*@*domain.name*** at a shell prompt. If the remote machine is **example.com** and your username on that machine is john, type **ssh john@example.com**.
>
> The **mount -o remount *file-system*** command remounts the named file system. For example, to remount the **/home** file system, the command is **mount -o remount /home**.
>
> To see the version of a currently installed package, use the **rpm -q *package*** command. It will return a result as follows: ***package-version-release***.

Note the words in bold italics above — username, domain.name, file-system, package, version and release. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

> Publican is a *DocBook* publishing system.

1.2. Pull-quote Conventions

Terminal output and source code listings are set off visually from the surrounding text.

Output sent to a terminal is set in **mono-spaced roman** and presented thus:

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

Source-code listings are also set in **mono-spaced roman** but add syntax highlighting as follows:

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;

public class ExClient
{
   public static void main(String args[])
       throws Exception
   {
      InitialContext iniCtx = new InitialContext();
      Object          ref    = iniCtx.lookup("EchoBean");
      EchoHome        home   = (EchoHome) ref;
      Echo            echo   = home.create();

      System.out.println("Created Echo");

      System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
   }
}
```

1.3. Notes and Warnings

Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.

Note

Notes are tips, shortcuts or alternative approaches to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.

Important

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring a box labeled 'Important' will not cause data loss but may cause irritation and frustration.

Warning

Warnings should not be ignored. Ignoring warnings will most likely cause data loss.

2. Getting Help and Giving Feedback

2.1. Do You Need Help?

If you experience difficulty with a procedure described in this documentation, visit the Red Hat Customer Portal at http://access.redhat.com. Through the customer portal, you can:

- search or browse through a knowledgebase of technical support articles about Red Hat products.
- submit a support case to Red Hat Global Support Services (GSS).
- access other product documentation.

Red Hat also hosts a large number of electronic mailing lists for discussion of Red Hat software and technology. You can find a list of publicly available mailing lists at https://www.redhat.com/mailman/listinfo. Click on the name of any mailing list to subscribe to that list or to access the list archives.

2.2. We Need Feedback!

If you find a typographical error in this manual, or if you have thought of a way to make this manual better, we would love to hear from you! Please submit a report in Bugzilla: http://bugzilla.redhat.com/ against the product **Red Hat Enterprise MRG.**

When submitting a bug report, be sure to mention the manual's identifier: *Management_Console_Installation_Guide*

If you have a suggestion for improving the documentation, try to be as specific as possible when describing it. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.

Chapter 1. Introduction to the MRG Management Console

1.1. About the MRG Management Console

The MRG Management Console provides a web-based graphical interface to manage your Red Hat Enterprise MRG deployment. It is based on the Cumin project.

MRG Messaging is built on the Qpid Management Framework (QMF). The MRG Management Console uses QMF to access data and functionality provided by the MRG Messaging broker (**qpidd**), MRG management system agent (**sesame**) and MRG Grid components.

The MRG Messaging broker is necessary for communication between the distributed MRG Grid components and Cumin. Sesame and the MRG Grid components must be installed on all nodes in the deployment.

Report a bug

1.2. About MRG Deployment Sizes

The MRG Management Console is designed to scale for deployments of MRG Messaging and MRG Grid. The following configurations indicate typical size and load characteristics for small, medium and large deployments.

Small

The default software configuration of the MRG Management Console is appropriate for small scale deployments. An example small scale deployment is:

- 64 nodes (each quad dual-core CPUs)
- 5 concurrent console users, accessing the console at 1 page view per second (peak)
- 10 job submitters, submitting 1 job per second concurrently (peak)
- 10 job completions per minute (sustained), 3 years of job history (1 million jobs)
- Ability to sustain peak rates for at least 5 minutes

Medium

An example medium scale deployment is:

- 500 nodes (each quad dual-core CPUs)
- 20 concurrent console users, accessing the console at 1 page view per second (peak)
- 20 job submitters, submitting 2 jobs per second concurrently (peak)
- 100 job completions per minute (sustained), 3 years of job history (10 million jobs)
- Ability to sustain peak rates for at least 5 minutes

Large

A large scale console is defined as a console supporting more than 5000 ***Execute Nodes*** and 100 concurrent users accessing the console at 1 page view per second during peak periods. There are several considerations when implementing a large scale console. Red Hat, Inc recommends that customers configure large scale MRG Management Console installations in cooperation with a Solutions Architect through Red Hat, Inc consulting.

Report a bug

Chapter 2. Preparing to Install MRG Management Console

2.1. About MRG Management Console Prerequisites

To install the MRG Management Console you need to have registered your system to the Red Hat Network. This table lists the Red Hat Enterprise MRG channels available on Red Hat Network for the MRG Management Console.

Table 2.1. Red Hat Network Channels for the MRG Management Console

Channel Name	Operating System	Architecture
MRG Grid v. 2	RHEL-5 Server	32-bit, 64-bit
MRG Grid v. 2	RHEL-6 Server	32-bit, 64-bit
MRG Management v. 2	RHEL-5 Server	32-bit, 64-bit
MRG Management v. 2	RHEL-6 Server	32-bit, 64-bit
MRG Messaging v. 2	RHEL-5 Server	32-bit, 64-bit
MRG Messaging v. 2	RHEL-6 Server	32-bit, 64-bit

Hardware Requirements

It is recommended that you have the following *minimum* hardware requirements before attempting to install the MRG Management Console:

- Intel Pentium IV or AMD Athlon class machine
- 512 MB RAM
- 10 GB disk space
- A network interface card

Report a bug

Chapter 3. Installing and Configuring the MRG Management Console

3.1. Quick Installation

3.1.1. About Quick Configuration of the MRG Management Console

The following procedures allow you to install Cumin, Sesame, the Qpid Messaging broker and HTCondor on a single machine with no changes to the packages' default configuration, then enable QMF communication on the nodes in the MRG Grid deployment. It is assumed that you have the required hardware and Red Hat Network channel subscriptions.

Report a bug

3.1.2. Install Cumin and the MRG Messaging Broker on a Single Host

1. On a machine running Red Hat Enterprise Linux 5 or 6, install the required packages for the MRG Management Console.

```
# yum install qpid-cpp-server cumin sesame
# export HISTIGNORE=*
# echo password | sudo -u qpidd /usr/sbin/saslpasswd2 -p -f
/var/lib/qpidd/qpidd.sasldb -u QPID cumin
# sed -i 's,^# brokers:.*,brokers: cumin/password@localhost:5672,'
/etc/cumin/cumin.conf
# unset HISTIGNORE
# /usr/sbin/cumin-database install
# /usr/sbin/cumin-admin add-user username
```

   A. Optionally, allow access to the user interface from other hosts on the network.

```
# sed -i 's,^# host:.*,host: 0.0.0.0,' /etc/cumin/cumin.conf
```

   B. Optionally, enable ***DIGEST-MD5*** authentication:

```
# yum install cyrus-sasl-md5
```

   C. Optionally, change the configuration of the interface for use with MRG Messaging (note that the MRG Grid interface will not be visible):

```
# yum install cumin-messaging
```

2. Start the MRG Messaging broker, Sesame and the MRG Management Console:

```
# /sbin/service qpidd start
# /sbin/service sesame start
# /sbin/service cumin start
```

3. Open a web browser and visit the URL http://localhost:45672. Log in as ***username***.

Report a bug

3.1.3. Enable Communication on HTCondor Nodes

1. Run the following commands on HTCondor nodes to enable QMF communication:

   ```
   # yum install condor-qmf sesame
   # echo "QMF_BROKER_HOST = broker_hostname" >
   /etc/condor/config.d/40QMF.config
   # sed -i 's,host=.*,host=broker_hostname,' /etc/sesame/sesame.conf
   ```

   A. Install Aviary on nodes running the HTCondor scheduler.

   ```
   # yum install condor-aviary
   ```

   > **Important**
   >
   > If *condor-aviary* is not installed, use of Aviary web services by Cumin must be disabled. Comments in **/etc/cumin/cumin.conf** explain how to disable Aviary in Cumin.

   B. Optionally, enable fair share editing on nodes running the HTCondor negotiator.

   ```
   # echo "NEGOTIATOR.ENABLE_RUNTIME_CONFIG = TRUE" >>
   /etc/condor/config.d/40QMF.config
   ```

   C. Optionally, adjust the collector update interval on the node running the collector.

   ```
   # echo "COLLECTOR_UPDATE_INTERVAL = 60" >>
   /etc/condor/config.d/40QMF.config
   ```

2. Restart the **sesame** and **condor** services.

   ```
   # /sbin/service sesame start
   # /sbin/service condor restart
   ```

Report a bug

3.2. Standard Installation

3.2.1. Install the MRG Management Console Broker

1. Install the MRG Management Console broker on any Red Hat Enterprise Linux host that is accessible over the network:

   ```
   # yum install qpid-cpp-server
   ```

   > **Note**
   >
   > If your host is running Red Hat Enterprise Linux 6, the *qpid-cpp-server* package is provided in the Red Hat Enterprise Linux 6 base channels. If your host is running Red Hat Enterprise Linux 5, this package is provided in the MRG Messaging channel.

Important

Password authentication must be available for use by the MRG Management Console.

2. It is recommended that you install the ***DIGEST-MD5*** password authentication mechanism. To do so, run the following command:

   ```
   # yum install cyrus-sasl-md5
   ```

3. The MRG Grid components will use ***ANONYMOUS*** authentication by default. Set the ***mech_list*** parameter in **/etc/sasl2/qpidd.conf** to enable these two mechanisms (all others will be disabled):

   ```
   mech_list: ANONYMOUS DIGEST-MD5
   ```

For more information on installing the MRG Messaging broker, see the *MRG Messaging Installation and Configuration Guide*.

Report a bug

3.2.2. Install Cumin

The Cumin installation requires a valid subscription to the Red Hat Network. Before you run the MRG Management Console for the first time, you will need to install the Cumin database.

1. **Install the Console**

   a. Install the MRG Management Console with the following **yum** command:

      ```
      # yum install cumin
      ```

   b. Install the recommended authentication mechanism:

      ```
      # yum install cyrus-sasl-md5
      ```

2. **Install the Cumin Database**

   a. Install the Cumin database with the following command:

      ```
      # /usr/sbin/cumin-database install
      ```

   b. If you are updating an existing Cumin installation, the following command will apply any necessary schema changes:

      ```
      # /usr/sbin/cumin-admin upgrade-schema
      ```

Report a bug

3.2.3. Install Sesame

Sesame is a MRG management system agent which monitors the statistics of every system on which it is installed. These system statistics are displayed in the MRG Management Console's **Inventory** page. It is recommended that Sesame is installed on every system in an MRG Grid deployment.

Use **yum** to install the *Sesame* package:

```
# yum install sesame
```

Report a bug

3.2.4. Install Grid Plug-ins

The HTCondor QMF plug-ins allow MRG Grid nodes to connect to a MRG Messaging broker.

- Install the QMF plug-ins on each node in the HTCondor pool:

  ```
  # yum install condor-qmf
  ```

Report a bug

Chapter 4. Configuring Cumin

4.1. Search for Parameters in the MRG Management Console

The MRG Management Console allows you to search for attributes in the various parameters of the MRG Grid implementation. For numeric parameters, the search feature offers additional query terms in the form of a set of operators that includes **=**, **<**, **<=**, **>**, **>=**.

> **Example 4.1. Use the MRG Management Console search feature**
>
> The following example shows the search feature used to return the idle slots in an MRG Grid implementation by querying the ***Activity*** attribute.
>
> **Procedure 4.1. Search for Idle Slots**
>
> 1. Click on the **Grid** panel in the MRG Management Console interface and select the **Slots** tab.
> 2. Click on the drop-down box and select the **Activity** attribute to select that column for the search query.
> 3. Type **Idle** into the search box and click the **Search** button.

Report a bug

4.2. Create SASL Credentials

Authentication credentials for the MRG Management Console must be created on the host running the MRG Messaging broker.

1. On the host, run the **saslpasswd2** command as the qpidd user:

```
$ sudo -u qpidd /usr/sbin/saslpasswd2 -f /var/lib/qpidd/qpidd.sasldb -u QPID
cumin
```

2. When prompted, create a password.
3. This command will create a ***cumin*** user in the SASL database.

Note

The qpidd user has permissions to read **/var/lib/qpidd/qpidd.sasldb**. If the ownership is wrong **/var/log/messages** will display a permission denied error.

Report a bug

4.3. Set the Broker Address and Authentication

The default configuration settings will connect the MRG Management Console without authentication to a MRG Messaging broker running on the same machine. You will need to change the default settings.

1. As the root user, open the **/etc/cumin/cumin.conf** file in a text editor and locate the ***brokers***

parameter, which is located under the ***[common]*** section. For example:

```
[common]
# database: dbname=cumin user=cumin host=localhost
brokers: cumin/oregano@alpha.example.com
```

The format of a broker address containing credentials is:

```
[<protocol>://]<username>/<password>@<target-host>[:<tcp-port>]
```

The optional ***tcp-port*** parameter will default to ***5672*** if not specified. The optional protocol value can be either ***amqp*** (the default) or ***amqps*** for SSL.

> **Note**
>
> The authentication information will be stored in plain text. However as permissions on this file are restricted the information will be secure provided users do not have root access.

2. The username value in this case *must be* ***cumin***, the user that was added during the SASL configuration.

   The password will be the password that you supplied when prompted by the **saslpasswd2** command. For example:

   ```
   brokers: cumin/oregano@alpha.example.com
   ```

By default the MRG Management Console will use the recommended ***DIGEST-MD5*** authentication mechanism if it has been installed. Refer to the ***sasl-mech-list*** parameter in the **/etc/cumin/cumin.conf** file if you wish to select other authentication mechanisms.

Report a bug

4.4. Configure Aviary Endpoints

Cumin uses the Aviary web services provided by MRG Grid to manage submissions. By default the MRG Management Console is configured to communicate with Aviary services using well-known ports on the local host. If the console is installed on a MRG Grid central manager which uses a default Aviary configuration, this section can be skipped. Otherwise, set the required parameters in **/etc/cumin/cumin.conf** to enable communication with Aviary.

Static Configuration

In **/etc/cumin/cumin.conf**, the ***aviary-job-servers*** and ***aviary-query-servers*** parameters in the ***[common]*** section are comma-separated lists of well-known URLs (or ***endpoints***) for the Aviary services. Modify these parameters to specify the lists of valid endpoints, for example:

```
[common]
aviary-job-servers: http://localhost:9123, http://alpha.example.com:9090
aviary-query-servers: http://localhost:9456, http://alpha.example.com:9091
```

Note

The **/etc/cumin/cumin.conf** file provides details on default values and short-hand notations that can be used when specifying endpoints.

Dynamic Configuration

Alternatively, the MRG Management Console uses the Aviary locator service to dynamically discover endpoints for services running in the pool. By default, the locator service is not enabled in MRG Grid or in the MRG Management Console. For information on enabling the locator service in MRG Grid, see the *MRG Grid Developer Guide*.

To enable use of the locator service by the MRG Management Console, the ***aviary-locator*** parameter must be explicitly set. In **/etc/cumin/cumin.conf**, uncomment the ***aviary-locator*** parameter and modify it if necessary:

```
[common]
aviary-locator: http://localhost:9000
```

When the locator service is used, specific values for ***aviary-job-servers*** and ***aviary-query-servers*** are ignored.

Important

If the locator service is enabled in MRG Grid, it must also be enabled in the MRG Management Console. If the locator service is disabled in MRG Grid, it must also be disabled in the MRG Management Console.

Report a bug

4.5. Configure SSL-Enabled Aviary

If the Aviary services in MRG Grid have been configured to use SSL, additional configuration changes must be made in **/etc/cumin/cumin.conf**:

1. Change the scheme from ***http*** to ***https*** for any Aviary service using SSL:

   ```
   [common]
   aviary-job-servers: https://localhost:9090
   ```

2. The default port for the Aviary locator service will change from ***9000*** to ***9443*** if SSL is enabled for the locator. Modify the ***aviary-locator*** parameter to reflect this change:

   ```
   [common]
   aviary-locator: https://localhost:9443
   ```

3. Set the ***aviary-key*** and ***aviary-cert*** parameters:

```
[common]
aviary-key: /etc/cumin/cumin.key
aviary-cert: /etc/cumin/cumin.crt
```

These parameters specify the full paths to a PEM formatted private key file and an associated PEM formatted certificate file which Cumin uses to communicate with the Aviary services. The certificate must also be installed as a client certificate on the host(s) on which Aviary services run. The security of these files is the responsibility of system administrators. They should be readable by the ***`cumin`*** user account.

4. Optionally, you can set the ***`aviary-root-cert`*** parameter.

   ```
   [common]
   aviary-root-cert: /etc/cumin/cumin-ca-bundle.crt
   aviary-domain-verify: True
   ```

   This is the full path to a PEM formatted file containing Certificate Authority (CA) certificates that the console uses to validate server certificates. The console will only validate server certificates if this parameter is set. The ***`aviary-domain-verify`*** parameter checks the hostname of the server against the server certificate, and is set to ***`True`*** by default.

Report a bug

4.6. Specify the Broker Address for Use of the Remote Configuration Feature

Cumin uses the remote configuration feature to augment inventory data and provide tag management facilities. The remote configuration feature (often referred to simply as '**`wallaby`**') consists of the Wallaby service, the wallaby command-line tool, and other tools and daemons that interact with the Wallaby service. For further information, see the Remote Configuration chapter in the *MRG Grid User Guide*.

By default, Cumin will use the first address specified in the ***`brokers`*** parameter as the address of the MRG Messaging broker for remote configuration. If that address is correct, this step can be skipped.

However, if the remote configuration feature is set up to use a different broker, the ***`wallaby-broker`*** parameter needs to be set accordingly, as described in the following procedure:

1. As the root user, open the **`/etc/cumin/cumin.conf`** file in a text editor and locate the ***`wallaby-broker`*** parameter, which is located under the ***`[common]`*** section. For example:

   ```
   [common]
   # database: dbname=cumin user=cumin host=localhost
   brokers: cumin/oregano@alpha.example.com
   # sasl-mech-list: [default, 'anonymous' or 'plain digest-md5' with usr/passw]
   # wallaby-broker: [default, first item in 'brokers' list]
   # wallaby-refresh: 60
   ```

2. Edit the ***`wallaby-broker`*** parameter to include the broker address you wish to use. For example:

   ```
   wallaby-broker: cumin/oregano@remoteconfig.example.com
   ```

3. You can also adjust the ***`wallaby-refresh`*** parameter to determine how often Cumin polls Wallaby. The default value is 60 seconds.

Report a bug

4.7. Set the Network Interface

The MRG Management Console is a web-based tool. You can use any internet browser to access the tool whether it is running on the local host or on a remote machine.

The web console is bound to the ***localhost*** network interface by default. This setting allows only local connections to be made. To make the MRG Management Console accessible to other machines on the network, the IP address of another network interface on the host needs to be specified in the configuration file.

1. Open the **/etc/cumin/cumin.conf** file and locate the ***[web]*** section.
2. On installation, the ***[web]*** section in the configuration file will have the following lines commented out. Remove the # symbol, then specify the IP address and port number to bind the web console to a different network interface:

```
[web]
host: 192.168.0.20
port: 1234
```

Note

> Setting the host parameter to ***0.0.0.0*** will make the web console bind to all local network interfaces.

Report a bug

4.8. Enable SSL for the MRG Management Console Web Server

The MRG Management Console web server can be configured to use SSL for secure communication. When SSL is enabled, the URL of the MRG Management Console will be in the format of ***https://host:port***.

1. Open the **/etc/cumin/cumin.conf** file and locate the ***[web]*** section.
2. Set the ***server-cert*** and ***server-key*** parameters to the paths of a valid certificate and a private key file respectively. For example:

```
[web]
server-cert: /etc/cumin/server.crt
server-key: /etc/cumin/server.key
```

   Both files must be in the PEM format, and readable by the ***cumin*** user. The security of these files is the reponsibility of the system administrator.

Report a bug

4.9. Set the MRG Management Console Persona

The default installation configures the MRG Management Console interface for use with MRG Grid. If the *cumin-messaging* package is installed, the console will instead be configured for use with MRG Messaging and the MRG Grid interface will not be visible. The MRG Management Console must be restarted after installing or removing this package for changes to take effect.

```
# yum install cumin-messaging
# /sbin/service cumin restart
```

Report a bug

4.10. Add and Remove Local Users

Users must be authenticated when logging into the MRG Management Console. This section explains how to create user accounts in the local database.

- Use this command to create an account in the database:

  ```
  # /usr/sbin/cumin-admin add-user user
  ```

  This will add a new user named ***user*** and prompt for a password. Using this form of the command ensures that passwords are not retained in the shell history.
- Use this command to remove a user account from the database:

  ```
  # /usr/sbin/cumin-admin remove-user user
  ```

Report a bug

4.11. Authentice Users with LDAP

The MRG Management Console can use LDAP directories for authentication provided that the LDAP servers allow simple bind. If a user cannot be found in the local database, the console will attempt to authenticate the user against a list of LDAP directories until a match is found or the list has been exhausted.

Configuring LDAP URLs

The MRG Management Console accepts standard LDAP URLs in the following format:

```
ldap://host:port/dn?attributes?scope?filter?extensions
```

However, the ***attributes*** and ***extensions*** values are ignored and can be omitted. This results in a URL in the following format:

```
ldap://host:port/dn??scope?filter
```

Additionally, ***scope*** will default to ***sub*** and ***filter*** will default to ***uid=username*** so that a URL can simply consist of host, port, and distinguished name (dn). If a filter is specified, the console will replace occurrences of ***%%s*** with ***username*** to construct the filter string.

Specify the list of LDAP URLs with the ***auth*** parameter in the ***[web]*** section of **/etc/cumin/cumin.conf**. Use semicolons (;) to separate multiple values.

```
[web]
auth: ldap://alpha.example.com/ou=users,dc=example,dc=com;
ldaps://beta.example.com:636/ou=people,dc=example,dc=com??one?mail=%%s
```

Using SSL with LDAP

Secure communication is recommended because passwords will be transmitted to LDAP servers during authentication. The documentation for your LDAP server will explain how to configure SSL for the server. To configure the MRG Management Console to use SSL with LDAP, set either or both of the following parameters in the ***[web]*** section of **`/etc/cumin/cumin.conf`**:

- ***ldap_tls_cacertfile***: This parameter specifies the full path to a PEM formatted certificate file which contains all the Certificate Authority (CA) certificates necessary to validate LDAP server certificates.
- ***ldap_tls_cacertdir***: This parameter specifies a directory containing PEM formatted CA certificates in individual files.

Setting either one of these parameters enables SSL with LDAP. If both parameters are set, the file specified by ***ldap_tls_cacertfile*** will be checked first. The ***cumin*** user must be able to read the files specified with these parameters. For example:

```
[web]
ldap_tls_cacertfile: /etc/cumin/ldapcert.crt
ldap_tls_cacertdir: /etc/cumin/
```

If SSL is enabled, the MRG Management Console will use the dynamic LDAP StartTLS mechanism over the specified port (default 389) for URLs beginning with the ***ldap*** scheme. The console also supports the ***ldaps*** scheme for dedicated SSL ports (default 636).

Report a bug

4.12. Enforce User Roles

Content in the MRG Management Console will be scoped by ***role*** if role enforcement is enabled. Every user account (including LDAP accounts) will be assigned the ***user*** role by default and can optionally be assigned the ***admin*** role. Accounts with the admin role have unrestricted access to all displays and functions. Accounts limited to the user role can access the **`Grid user`** display, and can manage their own jobs.

1. To enable role enforcement, edit the **`/etc/cumin/cumin.conf`** file and set the ***authorize*** parameter in the ***[web]*** section:

   ```
   [web]
   authorize: True
   ```

2. Use the **`cumin-admin`** command to manage roles.

   a. To assign a role, use the **`add-assignment`** parameter:

```
# /usr/sbin/cumin-admin add-assignment paul admin
User 'paul' is assigned to role 'admin'

# /usr/sbin/cumin-admin list-users
ID   Name                 Roles
----   -------------------- --------------------
1   paul                  user, admin
(1 user found)
```

b. To remove a role, use the **`remove-assignment`** parameter:

```
# /usr/sbin/cumin-admin remove-assignment paul admin
User 'paul' is no longer assigned to role 'admin'
```

3. Currently role information can only be set in the MRG Management Console database. Consequently, a special entry must be created in the database for LDAP accounts before their roles can be modified. To do so, use the **`cumin-admin external-user`** command:

```
# /usr/sbin/cumin-admin external-user john
External user 'john' is added
```

4. This creates an entry designating **`john`** as an external user. This account will still be authenticated via LDAP, but it can now be assigned a role:

```
# /usr/sbin/cumin-admin add-assignment john admin
User 'john' is assigned to role 'admin'
```

You can run the **`cumin-admin list-users`** command to check that the role has been correctly set:

```
# /usr/sbin/cumin-admin list-users
ID   Name                 Roles
----   -------------------- --------------------
2 * john                  user, admin
1   paul                  user
(2 users found)
(1 external user, indicated by *)
```

Note

It is not necessary to explicitly add external users to the MRG Management Console database unless their roles have to be modified.

Report a bug

4.13. Enable Kerberos Authentication in Cumin

Kerberos authentication can be used with the MRG Management Console for implementations using Red Hat Enterprise Linux 6 and above.

Procedure 4.2. Enable Kerberos Authentication

- To enable kerberos authentication, edit the **`/etc/cumin/cumin.conf`** file and set the ***auth*** parameter in the ***[web]*** section. Enter multiple values separated by semicolons if you also need LDAP authentication. Authentication mechanisms will be attempted in the order that they are given.

```
[web]
auth: kerb #
kerberos_realm: your_kerberos_realm
```

Currently, role information can only be set in the MRG Management Console database. Consequently, a special entry must be created in the database for kerberos accounts before their roles can be modified. To do so, use the **`cumin-admin external-user`** command.

Procedure 4.3. Create Kerberos Entry

1. Create an entry designating ***john*** as an external user.

   ```
   # /usr/sbin/cumin-admin external-user john
   ```

2. Assign the new account a role with the **`add-assignment`** command.

   ```
   # /usr/sbin/cumin-admin add-assignment john admin
   ```

   User ***john*** is able to be authenticated by Kerberos, and has been assigned to role ***admin***.

Report a bug

Chapter 5. Configuring Sesame

5.1. Set the Broker Address

Perform this configuration on all nodes where the *sesame* package is installed.

Open the **`/etc/sesame/sesame.conf`** file in a text editor and locate the ***`host`*** parameter. This parameter must be set to the hostname of the machine running the MRG Messaging broker:

```
host=example.com
```

The ***`port`*** parameter can also be set, although the default value should be correct for most configurations.

Report a bug

5.2. Configure Authentication

By default, ***`ANONYMOUS`*** authentication is permitted by the broker. If you wish to use ***`ANONYMOUS`*** authentication, this section can be skipped. However, for secure applications, the ***`DIGEST-MD5`*** mechanism is recommended. Use the following procedure to configure ***`DIGEST-MD5`*** authentication for Sesame.

1. Create credentials for all nodes running Sesame. Run the following command on the host where the broker is installed:

   ```
   $ sudo -u qpidd /usr/sbin/saslpasswd2 -f /var/lib/qpidd/qpidd.sasldb -u QPID
   sesame
   ```

2. This command will create a ***`sesame`*** user in the SASL database. For more information about the **`saslpasswd2`** command, refer to the *MRG Messaging Installation Guide*.
3. On each node where the *sesame* package is installed, open **`/etc/sesame/sesame.conf`** in a text editor and modify the following parameters.

   ```
   mech=DIGEST-MD5
   uid=sesame
   pwd=password
   ```

   Set the ***`mech`*** to ***`DIGEST-MD5`***. This value can be a space separated list if there are multiple supported mechanisms. Set ***`uid`*** to ***`sesame`*** and ***`pwd`*** to the password.

   > **Note**
   >
   > See configuration file comments on the ***`pwd-file`*** parameter if you wish to place the password in an external file.

4. Install the recommended authentication mechanism:

   ```
   # yum install cyrus-sasl-md5
   ```

Report a bug

Chapter 6. Configuring MRG Grid Plug-ins

6.1. About MRG Grid Plug-ins

The Grid plug-ins can be manually configured by creating a new configuration file on every MRG Grid node where the *condor-qmf* package has been installed. The remote configuration feature provides an alternative method to manage a MRG Grid deployment.

Report a bug

6.2. Set Broker Address and General Configuration

1. Create a new file in the **/etc/condor/config.d/** directory called **40QMF.config**:

   ```
   # cd /etc/condor/config.d/
   # touch 40QMF.config
   ```

2. To set the broker address on all nodes which are *not* running the MRG Messaging broker locally, add the following line to the **40QMF.config** file and specify the hostname of the machine running the broker:

   ```
   QMF_BROKER_HOST = <hostname>
   ```

3. To be able to edit fair-share in the MRG Management Console, edit the **40QMF.config** file on all nodes running the **condor_negotiator** to add the following line:

   ```
   NEGOTIATOR.ENABLE_RUNTIME_CONFIG = TRUE
   ```

   To enable runtime configuration of limit values it is vital that this line is present.

4. The sampling frequency of some graphs in the MRG Grid overview screens is related to how frequently the HTCondor collector sends updates. The default rate is fifteen minutes (900 seconds). This can be changed by adjusting the **COLLECTOR_UPDATE_INTERVAL** parameter.

   Do this by editing the new **40QMF.config** file on the node running the **condor_collector** to add the following line, with the desired value in seconds:

   ```
   COLLECTOR_UPDATE_INTERVAL = 60
   ```

5. Restart the **condor** service to pick up the changes (this command will also start the **condor** service if it is not already running):

   ```
   # /sbin/service condor restart
   ```

Report a bug

6.3. Configure Authentication

MRG Grid will authenticate to the MRG Messaging broker using the ***anonymous*** mechanism by default. If ***anonymous*** authentication is permitted by the broker, this step can be skipped. Otherwise use the following procedure:

1. On the host where the broker is installed, run the following command to create credentials for use

by all MRG Grid nodes:

```
$ sudo -u qpidd /usr/sbin/saslpasswd2 -f /var/lib/qpidd/qpidd.sasldb -u QPID
grid
```

2. When prompted, create a password. This command will create a ***grid*** user in the SASL database. For more information about the **saslpasswd2** command, refer to the *MRG Messaging Installation Guide*.

> **Note**
>
> The qpid user should be able to read **/var/lib/qpidd/qpidd.sasldb**. If the ownership is wrong **/var/log/messages** will display a permission denied error.

By default, ***ANONYMOUS*** authentication is permitted by the broker. If you wish to use ***ANONYMOUS*** authentication, this section can be skipped. However, for secure applications, the ***DIGEST-MD5*** mechanism is recommended. Use the following procedure to configure ***DIGEST-MD5*** authentication for MRG Grid nodes.

1. Create credentials for all MRG Grid nodes. Run the following command on the host where the broker is installed:

   ```
   $ sudo -u qpidd /usr/sbin/saslpasswd2 -f /var/lib/qpidd/qpidd.sasldb -u QPID
   grid
   ```

2. When prompted, create a password. This command will create a ***grid*** user in the SASL database. For more information about the **saslpasswd2** command, refer to the *MRG Messaging Installation Guide*.

3. Add the following lines to the **40QMF.config** file on every MRG Grid node where the *condor-qmf* package has been installed. Set the ***QMF_BROKER_AUTH_MECH*** parameter to ***DIGEST-MD5*** or another supported mechanism (this value can be a space separated list if there are multiple supported mechanisms):

   ```
   QMF_BROKER_AUTH_MECH = DIGEST-MD5
   QMF_BROKER_USERNAME = grid
   QMF_BROKER_PASSWORD_FILE = <path>
   CONFIGD_ARGS = --user grid --password <password>
   ```

   The ***QMF_BROKER_PASSWORD_FILE*** parameter specifies the path of a file containing the password for the ***grid*** user in plain text. The security of the password file is the responsibility of system administrators. The ***CONFIGD_ARGS*** parameter is required for **condor_configd** to connect to the broker with credentials, and is only necessary if remote configuration is used.

   > **Important**
   >
   > **condor_configd** uses a different mechanism for specifying passwords. At this time, it is necessary to specify the ***grid*** user's password in plain text in the **40QMF.config** file. The security of the configuration file is the responsibility of system administrators.

4. Install the recommended authentication mechanism:

```
# yum install cyrus-sasl-md5
```

Report a bug

6.4. Configure Grid Plug-ins with Remote Configuration

If the remote configuration feature is used to manage MRG Grid, several features must be applied to MRG Grid nodes to enable interaction with the MRG Management Console. More information on remote configuration features is available in the *MRG Grid User Guide* .

For a typical deployment, apply these features to all MRG Grid nodes:

- **`QMF`**
- **`ConsoleMaster`**
- **`ConsoleExecuteNode`**

Apply these additional features to the MRG Grid central manager:

- **`ConsoleNegotiator`**
- **`ConsoleCollector`**
- **`ConsoleScheduler`**
- **`JobServer`**
- **`QueryServer`** or **`SSLEnabledQueryServer`**

Report a bug

Chapter 7. Running

7.1. Start Services Manually

The **service** command can be used to manually start, stop, restart, or check the status of services on the local host.

1. Use these commands to start the following MRG services on the node(s) where they are installed:

   Starting the MRG Messaging broker:

   ```
   # /sbin/service qpidd start
   Starting Qpid AMQP daemon:                      [  OK  ]
   ```

   Starting Sesame:

   ```
   # /sbin/service sesame start
   Starting Sesame daemon:                      [  OK  ]
   ```

   Starting MRG Grid:

   ```
   # /sbin/service condor start
   Starting Condor daemons:                         [  OK  ]
   ```

   Starting the MRG Management Console database:

   ```
   # /usr/sbin/cumin-database start
   Starting postgresql service:                      [  OK  ]
   The database server is started
   ```

   Starting the MRG Management Console:

   ```
   # /sbin/service cumin start
   Starting Cumin:                       [  OK  ]
   ```

   Note

   The **/usr/sbin/cumin-database install** command must be run before the MRG Management Console can be started for the first time. .

2. After a configuration option has been changed, use the **/sbin/service** command to restart a running application:

   ```
   # /sbin/service cumin restart
   Stopping Cumin:                       [  OK  ]
   Starting Cumin:                       [  OK  ]

   # /sbin/service condor restart
   Stopping Condor daemons:                 [  OK  ]
   Starting Condor daemons:                 [  OK  ]
   ```

Report a bug

7.2. Start Services on System Boot

The MRG Management Console and associated services can be configured to start on system boot. Use the **chkconfig** command to check and set run levels for each service. This example configures services with default run levels 2, 3, 4, and 5.

```
# /sbin/chkconfig --list postgresql
postgresql      0:off 1:off 2:off 3:off 4:off 5:off 6:off
#
# /sbin/chkconfig postgresql on
# /sbin/chkconfig qpidd on
# /sbin/chkconfig sesame on
# /sbin/chkconfig condor on
# /sbin/chkconfig cumin on
```

Report a bug

7.3. Connect to the MRG Management Console

Open an internet browser and enter the web address (URL) for the MRG Management Console. The web address is the host and port where the Cumin service is running, for example http://localhost:45672/. The TCP port used by the MRG Management Console (default 45672) must be open for incoming traffic on the console host firewall to allow access from other hosts on the network.

Report a bug

7.4. About Logging

The MRG Management Console keeps log files in the **/var/log/cumin** directory. This directory will contain log files for the master script and each cumin-web or cumin-data process that is started as part of the cumin service.

Three log files are kept for each process and have the extensions **.log**, **.stderr** and **.stdout**. The **.log** file contains log entries from the running application. The **.stderr** and **.stdout** files contain redirected terminal output. Normally the **.stderr** and **.stdout** would be empty but they can contain error information. The master script makes an entry in the **master.log** file each time it starts or restarts another cumin process. If **/sbin/service** reports ***[FAILED]*** when cumin is started or if cumin does not seem to be running as expected, check these files for information.

A maximum log file size is enforced, and logs will be rolled over when they reach the maximum size. The maximum log file size and the number of rolled-over log files to archive can be set in the **/etc/cumin/cumin.conf** file with the ***log-max-mb*** and ***log-max-archives*** parameters.

Report a bug

Chapter 8. Advanced Configuration of the Management Console

8.1. Configuring the MRG Management Console for Medium Scale Deployment

8.1.1. About Advanced Configuration of the Management Console

Configuration considerations for deployments change as scale increases. This chapter describes advanced configuration of the MRG Management Console installation for medium scale deployments.

Report a bug

8.1.2. Run Multiple MRG Management Console Web Servers

In medium scale environments, it is often necessary to run multiple MRG Management Console web servers as the total number of page views per second increases. To ensure optimal performance, it is recommended that a single web server is used by no more than 20 to 30 simultaneous users. This section describes how to configure the MRG Management Console installation to run multiple web servers.

1. Create additional sections in **`/etc/cumin/cumin.conf`**.

   To add web servers, a new configuration section must be added to **`/etc/cumin/cumin.conf`** for each additional server. These sections have the same structure and default values as the standard ***[web]*** section with the exception of the ***log-file*** parameter. By default, each new server will log to a file in **`/var/log/cumin/`*****section_name*****`.log`**.

   Each new section must specify a unique value for ***port*** as each server binds to its own port. Adding the following lines to **`/etc/cumin/cumin.conf`** will add 3 new web servers to the configuration, ***web1***, ***web2*** and ***web3***; using default values for each server except ***port***. The default port for the ***web*** section is 45672.

   ```
   [web1]
   port: 45674

   [web2]
   port: 45675

   [web3]
   port: 45676
   ```

   The ***port*** values used above are chosen arbitrarily.

   The names of the sections created above must be added to the ***webs*** in the ***[master]*** section in order for the new web servers to run.

   ```
   [master]
   webs: web, web1, web2, web3
   ```

2. Check the configuration.

   After making the changes above, Cumin can be restarted. The **`/var/log/cumin/master.log`** file should contain entries for the new web servers.

```
# /sbin/service cumin restart
Stopping cumin:                                            [  OK  ]
Starting cumin:                                            [  OK  ]

# tail /var/log/cumin/master.log
...
20861 2011-04-01 12:09:45,560 INFO Starting: cumin-web --section=web --daemon
20861 2011-04-01 12:09:45,588 INFO Starting: cumin-web --section=web1 --
daemon
20861 2011-04-01 12:09:45,602 INFO Starting: cumin-web --section=web2 --
daemon
20861 2011-04-01 12:09:45,609 INFO Starting: cumin-web --section=web3 --
daemon
...
```

3. Access different servers.

   To visit a particular server, navigate using the appropriate ***port*** value. For example, on the machine where the MRG Management Console is installed, open an internet browser and navigate to http://localhost:45675/. This visits the ***[web2]*** server as configured above.

4. Troubleshoot.

   Make sure that the section names listed in the ***webs*** parameter of the ***[master]*** section are spelled correctly. Section naming errors can be identified by searching for **NoSectionError** in **/var/log/cumin/*.stderr**.

   If Cumin is running but cannot be accessed on a particular port as expected, make sure the port values specified in **/etc/cumin/cumin.conf** for each section are correct and that the ports are not used by any other application on the system.

   Whenever changes are made to **/etc/cumin/cumin.conf** the service must be restarted for the changes to take effect.

Note

The above instructions do not cover setting up a web server proxy; users must select a port manually. However, it is also possible to set up a proxy which handles load balancing automatically and allows users to visit a single URL rather than specific ports.

Report a bug

8.1.3. Increase the Default QMF Update Interval for MRG Grid Components

The default QMF update interval for MRG Grid components is 10 seconds. This interval affects how frequently MRG Grid notifies the MRG Management Console of changes in status. Increasing this interval for certain components can noticeably decrease load on the MRG Management Console.

Edit the **/etc/condor/config.d/40QMF.config** file to add the following recommended setting for a medium scale deployment:

```
STARTD.QMF_UPDATE_INTERVAL=30
```

Important

You can change the QMF update interval for any of the MRG Grid components, but in most cases you can retain the default configuration. If you change the **NEGOTIATOR.QMF_UPDATE_INTERVAL** value, ensure that it is less than or equal to the **NEGOTIATOR_INTERVAL** value (which defaults to 60 seconds). If either of these intervals are modified, check that this relationship still holds.

Report a bug

8.1.4. Tune the Cumin Database

max_connections

The ***max_connections*** parameter controls the number of simultaneous database connections allowed by the PostgreSQL server; the default value is 100. This value must be large enough to support the **cumin-web** and **cumin-data** processes that make up the MRG Management Console.

It is a good idea to check the value of this parameter if the MRG Management Console is configured to run multiple **cumin-web** instances or if other applications besides Cumin use the same PostgreSQL server.

The maximum number of concurrent connections needed by Cumin can be estimated with the following formula:

```
(cumin-web instances * 36) + (cumin-data instances) + 2
```

For a default Cumin configuration this number will be **43** but running multiple **cumin-web** instances will increase the number significantly.

If you receive the error message **OperationalError: FATAL: sorry, too many clients already** in the user interface, or contained in a **cumin** log file, this means that the available database connections were exhausted and a Cumin operation failed.

To change the allowed number of database connections, edit the **/var/lib/pgsql/data/postgresql.conf** file and set the ***max_connections*** parameter. The PostgreSQL server must be restarted for this change to take effect.

max_fsm_pages

The ***max_fsm_pages*** parameter in **/var/lib/pgsql/data/postgresql.conf** affects PostgreSQL's ability to reclaim free space. Free space will be reclaimed when the MRG Management Console runs the **VACUUM** command on the database (the vacuum interval can be set in **/etc/cumin/cumin.conf**). The default value for ***max_fsm_pages*** is 20,000. In medium scale deployments, it is recommended that ***max_fsm_pages*** be set to at least 64,000.

Important

> The following procedure is only applicable on a Red Hat Enterprise Linux 5 operating system, in which the PostgreSQL 8.1 database is in use. Red Hat Enterprise Linux 6 carries a later version of PostgreSQL, in which the ***`max_fsm_pages`*** parameter is no longer valid.

To set the ***`max_fsm_pages`*** parameter, use the following procedure:

1. Start an interactive PostgreSQL shell.

   ```
   $ psql -d cumin -U cumin -h localhost
   ```

2. Run the following command from the PostgreSQL prompt.

   ```
   cumin=# VACUUM ANALYZE VERBOSE;
   ```

   This will produce a large amount of output and takes several minutes to complete.
3. Edit the **`/var/lib/pgsql/data/postgresql.conf`** file and set the ***`max_fsm_pages`*** parameter to at least the indicated value from output of the previous command.
4. Restart the PostgreSQL service and perform this process again, repeating until PostgreSQL indicates that free space tracking is adequate:

   ```
   DETAIL:  A total of 25712 page slots are in use (including overhead).
   25712 page slots are required to track all free space.
   Current limits are:  32000 page slots, 1000 relations, using 292 KB.
   VACUUM
   ```

5. When PostgreSQL is restarted, restart Cumin for changes to take effect.

Report a bug

Chapter 9. Configuring the MRG Messaging Broker

9.1. Change the Update Interval

By default, the MRG Messaging broker will send updated information to the MRG Management Console every ten seconds. Increase the interval to receive fewer updates and reduce load on the broker or the network. Decrease the interval to receive more updates.

To change the update interval, open the ***/etc/qpidd.conf*** file in your preferred text editor and add the ***mgmt-pub-interval*** configuration option on the broker:

```
mgmt-pub-interval=30
```

Enter the required update interval in seconds.

Report a bug

9.2. Configure SSL

The MRG Messaging broker will always run with authentication checks turned on by default. Passwords will be sent to the MRG Messaging broker from the MRG Management Console in plain text. For greater security, SSL encryption can be used for communication between the MRG Management Console and the broker.

Warning

Cumin currently does not support the configuration setting **ssl-require-client-authentication=yes**. Setting this configuration option on the broker will prevent Cumin from connecting.

In the broker, SSL is provided through the **ssl.so** module. This module is installed and loaded by default in MRG Messaging. To enable the module, you need to specify the location of the database containing the certificate and key to use. This certificate database is created and managed by the Mozilla Network Security Services (NSS) **certutil** tool.

Use the following procedure to create a certificate database in **/var/lib/qpidd** and enable communication over SSL:

Important

The following procedure uses a self-signed certificate. In a secure environment, a certificate signed by a Certificate Authority (CA) is recommended. Refer to documentation on the **certutil** and **pk12util** commands for information on generating a Certificate Signing Request (CSR) and installing a signed certificate.

1. Create a file named **/var/lib/qpidd/*passwordfile*** to hold the certificate database password. This is a plain text file containing a single password. The file should be owned by the **qpidd** user and should not be readable by any other user. Ownership and permissions on the file can be set as follows:

```
# chown qpidd:qpidd /var/lib/qpidd/passwordfile
# chmod 600 /var/lib/qpidd/passwordfile
```

2. Create the database and insert a new certificate:

```
# cd /var/lib/qpidd
# sudo -u qpidd certutil -N -d . -f passwordfile
# sudo -u qpidd certutil -S -d . -f passwordfile -n nickname -s
"CN=nickname" -t "CT,," -x -z /usr/bin/certutil
```

3. Set the following options in the **/etc/qpidd.conf** configuration file:

```
ssl-cert-password-file=/var/lib/qpidd/passwordfile
ssl-cert-db=/var/lib/qpidd
ssl-cert-name=nickname
```

Note

The default port for SSL communication is ***5671***. This port can be changed by specifying the ***ssl-port*** option in the **/etc/qpidd.conf** file.

4. Install the *qpid-cpp-server-ssl* package:

```
# yum install qpid-cpp-server-ssl
```

5. Restart the broker.

```
# /sbin/service qpidd restart
```

After restarting, you can check the **/var/log/messages** file to quickly verify that the broker is listening for SSL connections. The message **Listening for SSL connections on TCP port 5671** indicates that SSL communication has been successfully configured.

6. Clients can now communicate with the broker using a URL specifying the ***amqps*** protocol and the SSL port number, for example amqps://localhost:5671.

Important

The ***brokers*** parameter in **/etc/cumin/cumin.conf** must be changed to specify the amqps protocol and the SSL port number, and enable the MRG Management Console to restart using SSL.

For more information on setting up SSL encryption, refer to the *MRG Messaging User Guide*.

Report a bug

9.3. Add Credentials to Optional Broker ACLs for MRG Services

The MRG Messaging broker can be configured to use an access control list (ACL). If an ACL has been created for the MRG Messaging broker, ensure that any SASL users that have been created for Cumin, Sesame and MRG Grid are handled in the ACL. Note that if MRG Grid or Sesame is using ***anonymous*** authentication, the ***anonymous@qpid*** user must also be added.

For example, these additions to an ACL file grant unrestricted access to the users **cumin**, **grid**, and **sesame**:

```
acl allow cumin@QPID all all
acl allow grid@QPID all all
acl allow sesame@QPID all all
```

For a full discussion of ACLs, see the *MRG Messaging User Guide* sections on security and authorization.

Report a bug

Chapter 10. Frequently Asked Questions

10.1. Frequently Asked Questions

A collection of frequency asked questions and troubleshooting advice.

Q: If I uninstall, reinstall or update the Cumin software will my database be lost?

A: No, the data in the database will persist. Even an uninstall, reinstall, or update of PostgreSQL should not affect your data. However, you're advised to back up the database prior to any such operations (more information on backup can be found in the PostgreSQL documentation).

Q: So what if I want to create a fresh database?

A: To discard your data, the database must be destroyed and recreated. Optionally, you can preserve the user account data during this procedure.

To backup your user account data:

```
# /usr/sbin/cumin-admin export-users my_users
```

Then destroy the old database and create a new one:

Warning

This command will cause you to lose all data previously stored in the database. Use only with extreme caution.

```
# /usr/sbin/cumin-database drop
# /usr/sbin/cumin-database create
```

To restore your user account data:

```
# /usr/sbin/cumin-admin import-users my_users
```

Q: Help! My database is corrupted! What do I do now?

A: If the database is completely corrupted, the easiest way to fix the problem is to destroy the old database, and create a new one as described above.

Q: Will I ever be required to recreate my database as part of a software upgrade?

A: Occasionally, new features in Cumin require changes to the database schema. If this is the case, the Release Notes will inform you that the database must be recreated for use with the new version of software. If practical, additional instructions or facilities will be included to help with the transition.

Q: If I have to recreate my database, what will I actually lose?

A: Presently Cumin stores 24 hours of sample data for calculating statistics along with user account data and information about agents and objects it discovers through QMF. Cumin will dynamically rediscover agents and objects while it runs, so this type of data is not really lost.

User account data will be lost but can be restored as described above, this is assuming it has previously been exported with **`/usr/sbin/cumin-admin`**. Sample data from the last 24 hours will be lost, affecting some statistics and charts displayed by Cumin.

Q: **How can I make the graph labeled `Grid - Overview, Host info` update more `frequently`?**

A: The data comes from the Collector, controlled by the **`COLLECTOR_UPDATE_INTERVAL`**. The default value is 900 seconds (15 minutes). For more frequent updates, set it to a smaller value, such as 30, on the nodes where the **`condor_collector`** is running. This can be done in **`/etc/condor/config.d/40QMF.config`**.

Report a bug

Chapter 11. More Information

11.1. Generating Certificates with OpenSSL

11.1.1. Reference of Certificates

This reference for creating and managing certificates with the **openssl** command assumes familiarity with SSL. For more background information on SSL refer to the OpenSSL documentation at www.openssl.org.

> **Important**
>
> It is recommended that only certificates signed by an authentic Certificate Authority (CA) are used for secure systems. Instructions in this section for generating self-signed certificates are meant to facilitate test and development activities or evaluation of software while waiting for a certificate from an authentic CA.

Generating Certificates

Procedure 11.1. Create a Private Key

- Use this command to generate a 1024-bit RSA private key with file encryption. If the key file is encrypted, the password will be needed every time an application accesses the private key.

  ```
  # openssl genrsa -des3 -out mykey.pem 1024
  ```

  Use this command to generate a key without file encryption:

  ```
  # openssl genrsa  -out mykey.pem 1024
  ```

Procedure 11.2. Create a Self-Signed Certificate

Each of the following commands generates a new private key and a *self-signed* certificate, which acts as its own CA and does not need additional signatures. This certificate expires one week from the time it is generated.

1. The ***nodes*** option causes the key to be stored without encryption. OpenSSL will prompt for values needed to create the certificate.

   ```
   # openssl req -x509 -nodes -days 7 -newkey rsa:1024 -keyout mykey.pem -out
   mycert.pem
   ```

2. The ***subj*** option can be used to specify values and avoid interactive prompts, for example:

   ```
   # openssl req -x509 -nodes -days 7 -subj
   '/C=US/ST=NC/L=Raleigh/CN=www.redhat.com' -newkey rsa:1024 -keyout mykey.pem
   -out mycert.pem
   ```

3. The ***new*** and ***key*** options generate a certificate using an existing key instead of generating a new one.

   ```
   # openssl req -x509 -nodes -days 7 -new -key mykey.pem -out mycert.pem
   ```

Create a Certificate Signing Request

To generate a certificate and have it signed by a Certificate Authority (CA), you need to generate a certificate signing request (CSR):

```
# openssl req -new -key mykey.pem -out myreq.pem
```

The certificate signing request can now be sent to an authentic Certificate Authority for signing and a valid signed certificate will be returned. The exact procedure to send the CSR and receive the signed certificate depend on the particular Certificate Authority you use.

Create Your Own Certificate Authority

You can create your own Certificate Authority and use it to sign certificate requests. If the Certificate Authority is added as a trusted authority on a system, any certificates signed by the Certificate Authority will be valid on that system. This option is useful if a large number of certificates are needed temporarily.

1. Create a self-signed certificate for the CA, as described in Procedure 11.2, “Create a Self-Signed Certificate”.
2. OpenSSL needs the following files set up for the CA to sign certificates. On a Red Hat Enterprise Linux system with a fresh OpenSSL installation using a default configuration, set up the following files:
    a. Set the path for the CA certificate file as **`/etc/pki/CA/cacert.pem`**.
    b. Set the path for the CA private key file as **`/etc/pki/CA/private/cakey.pem`**.
    c. Create a zero-length index file at **`/etc/pki/CA/index.txt`**.
    d. Create a file containing an initial serial number (for example, 01) at **`/etc/pki/CA/serial`**.
    e. The following steps must be performed on RHEL 5:
        a. Create the directory where new certificates will be stored: **`/etc/pki/CA/newcerts`**.
        b. Change to the certificate directory: **`cd /etc/pki/tls/certs`**.
3. The following command signs a CSR using the CA:

    ```
    # openssl ca -notext -out mynewcert.pem -infiles myreq.pem
    ```

Install a Certificate

1. For OpenSSL to recognize a certificate, a hash-based symbolic link must be generated in the **`certs`** directory. **`/etc/pki/tls`** is the parent of the **`certs`** directory in Red Hat Enterprise Linux's version of OpenSSL. Use the **`version`** command to check the parent directory:

    ```
    # openssl version -d
    OPENSSLDIR: "/etc/pki/tls"
    ```

2. Create the required symbolic link for a certificate using the following command:

    ```
    # ln -s certfile `openssl x509 -noout -hash -in certfile`.0
    ```

    It is possible for more than one certificate to have the same hash value. If this is the case, change the suffix on the link name to a higher number. For example:

    ```
    # ln -s certfile `openssl x509 -noout -hash -in certfile`.4
    ```

Examine Values in a Certificate

The content of a certificate can be seen in plain text with this command:

```
# openssl x509 -text -in mycert.pem
```

Exporting a Certificate from NSS into PEM Format

Certificates stored in an NSS certificate database can be exported and converted to PEM format in several ways:

- This command exports a certificate with a specified nickname from an NSS database:

  ```
  # certutil -d . -L -n "Some Cert" -a > somecert.pem
  ```

- These commands can be used together to export certificates and private keys from an NSS database and convert them to PEM format. They produce a file containing the client certificate, the certificate of its CA, and the private key.

  ```
  # pk12util -d . -n "Some Cert" -o somecert.pk12
  # openssl pkcs12 -in somecert.pk12 -out tmckay.pem
  ```

  See documentation for the **`openssl pkcs12`** command for options that limit the content of the PEM output file.

Report a bug

Revision History

Revision 3.1-3	**Fri Mar 01 2013**	**David Ryan**

Preparing in Pressgang CCMS.

Revision 3.0-6	**Thu Feb 14 2013**	**David Ryan**

Migrated book to Pressgang CCMS.

Revision 3.0-5	**Fri Nov 01 2012**	**David Ryan**

Minor bug fixes.

Revision 3.0-4	**Fri Oct 19 2012**	**David Ryan**

Minor bug fixes.

Revision 3.0-3	**Fri Oct 19 2012**	**David Ryan**

BZ#867625 - Code example bug fix.

Revision 3.0-2	**Mon Oct 15 2012**	**Cheryn Tan**

Prepared for publishing (MRG 2.2.1).

Revision 3.0-0	**Fri Sep 14 2012**	**Cheryn Tan**

Prepared for publishing (MRG 2.2).

Revision 2-21	**Fri Aug 10 2012**	**Cheryn Tan**

Rebuild for Publican 3.0

Revision 2-20	**Mon Aug 6 2012**	**Cheryn Tan**

Updated details on Red Hat Network channels for the management console

Revision 2-18	**Wed Jul 18 2012**	**Cheryn Tan**

Added new overview diagram and rewrote introduction to the management console

Revision 2-17	**Wed Jul 17 2012**	**Cheryn Tan**

Minor spelling and grammar edits from QE review

Revision 2-16	**Wed Jun 27 2012**	**Cheryn Tan**

BZ#834474 - Edited procedure on starting services automatically

Revision 2-15	**Fri Jun 22 2012**	**Cheryn Tan**

BZ#819946 - Edited security appendix
Removed content irrelevant for 2.2 release

Revision 2-14	**Tue Jun 12 2012**	**Cheryn Tan**

BZ#819946 - Added security appendix

Revision 2-13	**Mon Jun 4 2012**	**Cheryn Tan**

BZ#821063 - Added section on configuring SSL for Cumin web server
BZ#738777 - Rearranged LDAP sections on user authentication
BZ#754224 - Added admonition about using self-signed certificates for the Messaging broker

Revision 2-12 **Mon May 21 2012** **Cheryn Tan**

BZ#733707 - Added configuration parameters for Aviary using SSL
BZ#738777 - Documented using SSL with LDAP
BZ#761381 - Added Messaging channels to list of RHN requirements
BZ#750273 - Removed references to wallaby tool

Revision 2-11 **Fri May 4 2012** **Cheryn Tan**

BZ#818691 - Limited scope of ENABLE_RUNTIME_CONFIG
BZ#754228 - Minor edits to configuring authentication for Grid plug-ins

Revision 2-10 **Wed May 2 2012** **Cheryn Tan**

BZ#750273 - Added section on remote configuration of Grid plugins
BZ#811230 - Documented user and admin role enforcement
BZ#760759 - Added overview diagram of management console
BZ#754227 - Removed reference to modifying mech_list from Quick Start section
BZ#754228 - Edited authentication configuration for Grid plug-ins
BZ#738777 - Documented LDAP authentication for cumin

Revision 2-9 **Wed Apr 4 2012** **Cheryn Tan**

BZ#761381 - Incorporated suggestions from QE review
BZ#754227 - Divided document into parts, added quick-start chapter

Revision 2-8 **Mon Apr 2 2012** **Cheryn Tan**

BZ#806696, BZ#806697, BZ#806698 - Docs QE reviews
BZ#803766 - Changed job server configuration to match new condor plugin defaults
BZ#802918 - Added cumin-admin upgrade-schema command
BZ#798805 - Edited sections on persona value
BZ#790151 - Edited service, cumin-admin and cumin-database commands
BZ#754228 - Changed references of PLAIN authentication to DIGEST-MD5

Revision 2-7 **Tue Feb 28 2012** **Tim Hildred**

Updated configuration file for new publication tool

Revision 2-4 **Mon Jan 16 2012** **Cheryn Tan**

Fixed typos, rearranged content on tuning Cumin database

Revision 2-3 **Thu Jan 12 2012** **Cheryn Tan**

BZ#754224 - Moved content on setting up SSL and configuring ACLs to new appendix
BZ#753867 - Reverted to original hardware requirements
BZ#754223 - Rewrote firstrun instructions

Revision 2-2 **Fri Jan 6 2012** **Cheryn Tan**

BZ#753867 - Edited hardware requirements
BZ#768192 - Added appendix on max_fsm_pages

Revision 2-1 **Thu Jan 5 2012** **Cheryn Tan**

BZ#768192 - Added PostgreSQL max_connections for Cumin users
BZ#754225 - Link between wallaby and remote configuration feature

Revision 2-0 **Tue Dec 6 2011** **Alison Young**

Prepared for publishing

Revision 1-18 **Tue Nov 29 2011** **Alison Young**

BZ#731801 - minor update

Revision 1-17 **Thu Nov 24 2011** **Alison Young**

BZ#752912 - comment 23

Revision 1-16 **Nov 18 2011** **Alison Young**

BZ#752912 - addressed comments 14 - 21

Revision 1-15 **Thur Nov 17 2011** **Alison Young**

BZ#752406 - change RHEL versions
BZ#752912 - addressed comments 3 - 11

Revision 1-14 **Mon Nov 14 2011** **Alison Young**

BZ#629912 - FAQ clarification
BZ#752912 - Tech Review of 2.1 MCIG

Revision 1-11 **Mon Nov 07 2011** **Alison Young**

BZ#750820 - Instructions for configuring Sesame authentication are wrong and/or missing

Revision 1-10 **Mon Oct 24 2011** **Alison Young**

BZ#738793 - updates from review

Revision 1-9 **Fri Oct 21 2011** **Alison Young**

BZ#733683 - Added information on wallaby-broker configuration parameter
BZ#738793 - updates from review

Revision 1-8 **Thu Oct 20 2011** **Alison Young**

BZ#731813 - authentication as cumin user is necessary for job ops
BZ#738793 - updates from review

Revision 1-7 **Mon Oct 18 2011** **Alison Young**

BZ#629912 - How to Adjust update rate on slot utilization graph in Cumin
BZ#706096 - Remove content on configuration file ownership
BZ#731799 - "one of two ways" to configure the job server clarification
BZ#731801 - Review config instructions for jobserver
BZ#738785 - Missing preposition in chapter 5
BZ#738793 - Restructure review updates

Revision 1-6 **Mon Oct 17 2011** **Alison Young**

BZ#738793 - continued restructure

Revision 1-5 **Fri Oct 14 2011** **Alison Young**

BZ#738793 - commenced restructure

Revision 1-4 **Thu Oct 13 2011** **Alison Young**

BZ#731811 - cumin-database install and cumin-admin add-user must be run from root shell
BZ#737637 - Extra quotes in section 4.1

BZ#738792 - Name of the qpidd user in Note in section 2.1 is given as 'qpid'

Revision 1-3	Wed Sep 07 2011	Alison Young

Prepared for publishing

Revision 1-1	Wed Sep 07 2011	Alison Young

BZ#735358 - Update for adding cumin and grid to sasldb

Revision 1-0	Thu Jun 23 2011	Alison Young

Prepared for publishing

Revision 0.1-5	Tue May 31 2011	Alison Young

Rebuilt as some changes missing from previous build.

Revision 0.1-4	Mon May 30 2011	Alison Young

Technical review fixes
BZ#674834 - treatment of data on uninstall/upgrade/reinstall
BZ#705828 - Sesame installation updates
BZ#706182 - configuration parameter settings for Job Server
BZ#706446 - RHEL-6 Server channel missing from table 2.1

Revision 0.1-3	Thu Apr 07 2011	Alison Young

BZ#692227 - setting sasl_mech_list parameter in cumin.conf
BZ#696223 - Changed section 2.1 default MRG Messaging set up has changed

Revision 0.1-2	Thu Apr 07 2011	Alison Young

BZ#681283 - Scale Documentation (2.x)
BZ#689785 - Change default QMF update interval, special config for submissions
BZ#690453 - setting the 'persona' value for console specialization
BZ#692983 - subsection on logging to Chapter 3

Revision 0.1-1	Tue Apr 05 2011	Alison Young

BZ#687872- Need instructions for anonymous@QPID plugin authentication
added update from v1.3 for BZ#634932 - Runtime Grid config setting

Revision 0.1-0	Tue Feb 22 2011	Alison Young

Fork from 1.3