w: www.mariadb.com

e: info@mariadb.com

MariaDB Enterprise Cluster

Getting Started Guide

w: www.mariadb.com

e: info@mariadb.com

Table of Contents

w: www.mariadb.com

e: info@mariadb.com

1 What is MariaDB Enterprise Cluster?

MariaDB Enterprise Cluster is composed of several components including MariaDB Manager, which is a set of management tools and an API with which you can easily provision, monitor, and manage a highly available MariaDB Galera Cluster for multi-master, synchronous replication. Galera is a powerful technology that can eliminate single points of failure for your database infrastructure, but it is relatively new and can be a challenge to configure for administrators who aren't familiar with it.

MariaDB Enterprise Cluster removes the guesswork from initially provisioning and configuring a MariaDB Galera Cluster. It handles monitoring the health of a cluster, and performing basic management tasks such as starting and stopping nodes, isolating and rejoining nodes, and performing backups and restores.

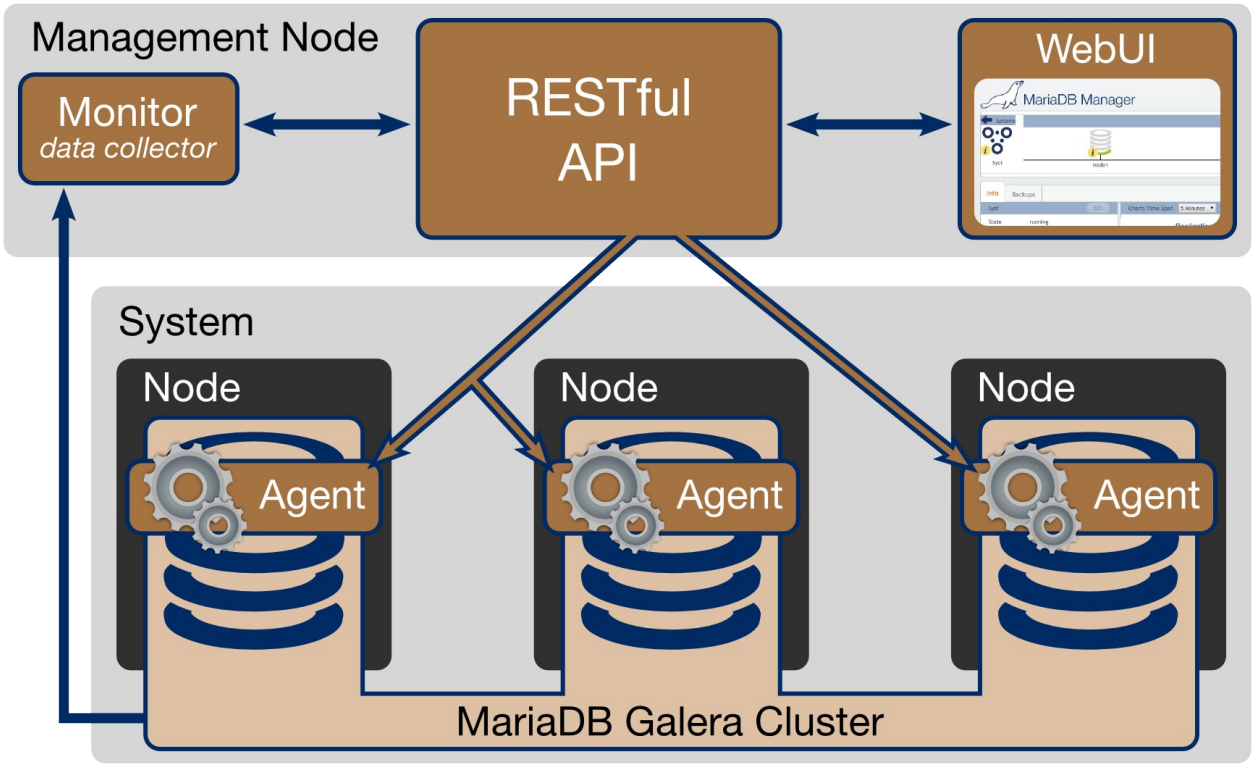
MariaDB Enterprise Cluster includes the following integrated components:

- **MariaDB Manager** — an open source cluster monitoring and management solution.
- **MariaDB Galera Cluster** — an open source, synchronous, multi-master MariaDB cluster using Galera.
- Optional **Webyog MONyog** — a commercial monitoring solution. This tool is included as part of MariaDB Enterprise Cluster and may be installed if needed.
- **Support** — technical support subscription from SkySQL.

The MariaDB Manager component includes:

- An **API** — the central hub that provides the provisioning, monitoring, and management capabilities through a RESTful API.
- A **Monitor** — a data collection engine installed on the management node, that connects to the database nodes to gather operational statistics and then stores and analyzes those statistics and delivers data to the API.
- A **WebUI** — a graphical user interface (GUI) to the monitoring and administration features of MariaDB Manager, installed on the management node.
- An **Agent** — a set of scripts, installed on each node, that enable MariaDB Manager to interact with the nodes through the API.

The API engine provisions and manages database nodes through the agent installed on each node. This agent then receives instructions from the API engine. Using the agent, MariaDB Manager is capable of installing, provisioning, and managing a MariaDB Galera Cluster database node. The following graphic illustrates how all of these parts work together:



This architecture allows MariaDB Manager to control a highly available MariaDB database cluster built using the Galera multi-master, synchronous clustering tools. To control the cluster, MariaDB Manager defines several abstractions which simplify the process and allow you to manage such a cluster through either the WebUI or the API.

Nodes

The server resources which host either the MariaDB Manager itself, or the individual database instances to be clustered together are called *nodes*. A node may be a physical server or blade, a virtual server, or a cloud-based server instance such as an Amazon Machine Instance (AMI). Nodes are managed by the MariaDB Manager in concert to deliver a clustered database resource.

Systems

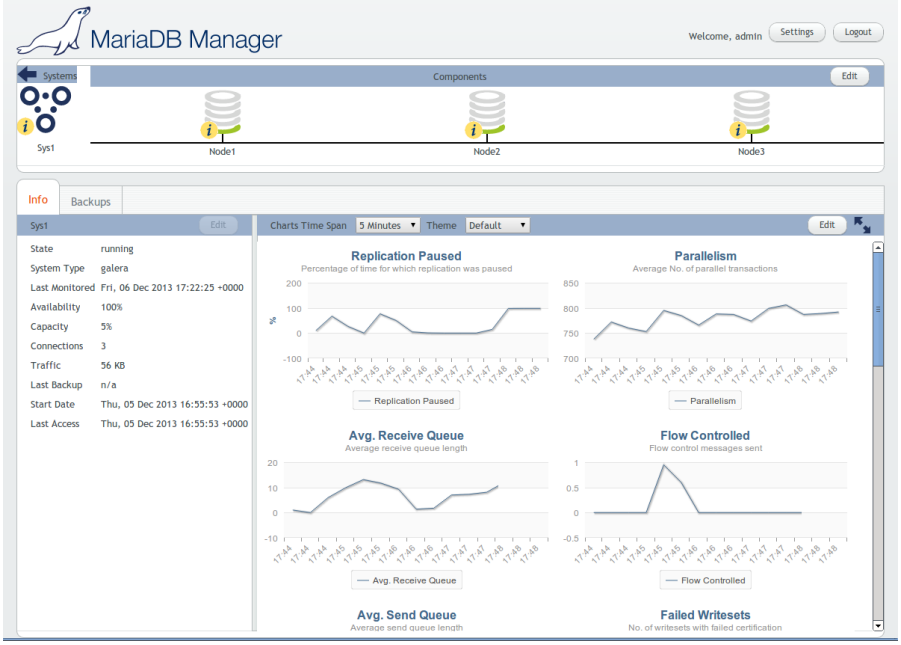
A *system* is a collection of nodes which MariaDB Manager groups together as a common resource. For example a Galera-based MariaDB database cluster is a system. The management node itself on which MariaDB Manager runs is another system. MariaDB Manager can provision, monitor, and manage multiple systems at once, through both the API and WebUI.

You will see these terms used in describing the objects on which you can operate using the tools built into MariaDB Manager, below.

In addition to the MariaDB Manager, MariaDB Enterprise Cluster also includes the optional Webyog MONyog monitoring tool, and 24x7 technical support, which together deliver a complete solution for easily deploying highly available MariaDB database clusters in production environments.

2 Introducing MariaDB Manager

MariaDB Manager uses the API to interact with servers running MariaDB Galera Cluster. Included with MariaDB Manager is a web-based user interface (WebUI). This is the default interface and is what will be used in the examples in this document. With MariaDB Manager you can install multiple systems containing sets of replication servers or nodes running MariaDB Galera Cluster. It can also provision MariaDB Galera Cluster on the nodes within systems — making installation and configuration much easier. The MariaDB Manager WebUI facilitates common management tasks for MariaDB Galera Cluster nodes, letting you view information on each system and each node within a system, setup replication between servers, stop servers, isolate servers from others to make clean backups, and bring stopped servers back on-line.



It is worth noting that other interfaces or applications can be built to interact directly with the API to do all of these tasks. The WebUI is simply the default.

2.1 Installing MariaDB Manager

We recommend that you first install MariaDB Manager on its own server, and then that server will handle the installation of agents, and the provisioning of MariaDB Galera Cluster on all other nodes. This initial installation is managed in the usual way by the `yum` package manager, with a supplied repository. You just need to make sure the servers you are using meet the system requirements.

2.1.1 System Requirements

There are only a few prerequisites for installing MariaDB Manager. The management server should be running a 64-bit version of CentOS 6.x, have at least a dual-core CPU, at least 2GB of RAM, and at least 1GB of disk space for storing logs and other data.

MariaDB Manager needs root access only when being set up. Once this main server is in place, you can proceed to installing your replication nodes.

Also, in order to provision replication nodes, the `openssh-clients` and `openssh-server` packages must be install on every node. After installing them, enable and start the `sshd` daemon with the following commands (run as root or with `sudo`):

```
chkconfig sshd on
service sshd start
```

2.1.2 Installing from the YUM Repository

Yum requires a configuration file to provide repository paths and parameters for the installation process. You can create this file under `/etc/yum.repos.d/` (i.e., `/etc/yum.repos.d/MariaDB-Manager.repo`), either with a simple text editor or by downloading a ready-made file. Following is an example of the file for illustration purposes, but we recommend that you download the latest version directly from the mariadb.com website, as its contents may have changed.

```
[MariaDB-Manager]
name=MariaDB-Manager
baseurl=http://user:pwd@code.mariadb.com/MariaDB-Manager/R1.0.0/repo/
enabled=1
gpgcheck=false
```

You can use the curl program to easily download this file and place it in its proper location. To do so, run the following as the root user on your management server (replacing `user:pwd` with your own username and password). The command is split over three lines for clarity and to avoid wrapping, but it can all be on one line if the slashes (`\`) at the end of the first two lines are removed (they are there to let the system shell know the command continues on the following line).

```
curl --user user:pwd \
http://code.mariadb.com/MariaDB-Manager/R1.0.0/MariaDB-Manager.repo \
> /etc/yum.repos.d/MariaDB-Manager.repo
```

Once the repository configuration file is in place, you will need to edit it to add your username (`user`) and password (`pwd`) to the file. The file from the MariaDB website has placeholders for our username and password on the baseurl line. You should have received these credentials as part of your welcome package when you purchased MariaDB Enterprise Cluster. Open the file in a text editor and replace the user:pwd

text with your username and password, separated by a colon (:). Save and close the configuration file, and then run the following command as the root user:

```
yum install MariaDB-Manager
```

Your system will be checked for any dependencies and yum will then present you with a list of packages to install. Unless the package manager asks to uninstall something that you require for other services which are running on the management server, or it proposes to install software package upgrades that might cause problems, it is generally safe to accept the list of recommendations. Once yum has finished installing everything, you will use MariaDB Manager to configure your setup, and to install and connect to your replication servers.

If your username or password contains characters that appear elsewhere in the baseurl you need to encode them using the following table:

Character	Encoding
@	%40
/	%2f
:	%38
%	%25

For example, if your username is `tomjones` and your password is `wh@ts:new` you would execute the previous curl example as follows:

```
curl --user tomjones:wh%40ts%38new \
http://code.mariadb.com/MariaDB-Manager/R1.0.0/MariaDB-Manager.repo \
> /etc/yum.repos.d/MariaDB-Manager.repo
```

2.1.3 Ports

MariaDB Manager does not handle firewall configuration on nodes or the network those nodes are on. For reference, the following ports and protocols need to be open at least between nodes (and in the case of the management node, you will access the MariaDB Manager WebUI on port 8080):

On the management node where you have installed MariaDB Manager and the API components:

- 22 (SSH)
- 80 (HTTP)
- 8080 (HTTP*)
- ICMP

On the database nodes:

- 22 (SSH)

- 3306 (MariaDB default port)
- 4444 (XtraBackup state transfer port)
- 4567 (XtraBackup group communication port)
- 4568 (XtraBackup incremental state transfer port)
- ICMP

2.1.4 Connecting securely to the MariaDB Manager

While the connections between the MariaDB Manager and database nodes are secure, the connection between your browser (client) and the MariaDB Manager is not secure by default. This is a common problem for monitoring tools such as MariaDB Manager; many other popular database monitoring solutions have the same issue. If you are concerned about securing the link between your browser and the MariaDB Manager, you can easily establish an SSH tunnel for this http traffic by forwarding a free TCP port on your client system to the MariaDB Manager node's port 80. The following command will accomplish this port forwarding, creating an encrypted tunnel between your client browser and the MariaDB Manager:

```
ssh -f -L <user>@<mariadb-manager-node> -L 8080:<mariadb-manager-node>:80 -N
```

For **<user>**, substitute a valid login user name on the node running the MariaDB Manager. For **<mariadb-manager-node>** substitute the host name or the IP address of the node running the MariaDB Manager. You can choose any free port on your client system, 8080 is just an example.

Once the tunnel is set up, visit: http://localhost:8080/ in your browser to access the MariaDB Manager. This port is tunneled and encrypted to port 80 on the MariaDB Manager's node, and you will be accessing the MariaDB Manager's interface securely.

2.2 Configuring Systems

Configuring a highly available database cluster with MariaDB Manager is straightforward. You log in to a web interface, create empty systems (clusters), and then add database nodes within each system. The WebUI calls the API to install and provision MariaDB and Galera software on the nodes, and automatically configure the nodes to work together as a highly available database cluster. MariaDB Manager automates this time consuming and potentially error-prone manual process, making Galera configuration quick and easy.

2.2.1 The Initial Start

After installing MariaDB Manager on a server, the web-based user interface will be running and you can access it using a web browser. For the address, enter the domain name or IP address of the server in the browser address box followed by a colon, then the port number (8080), a slash (/), and then MariaDBManager. It will look like this:

```
http://xx.xx.xx.xx:8080/MariaDBManager
```

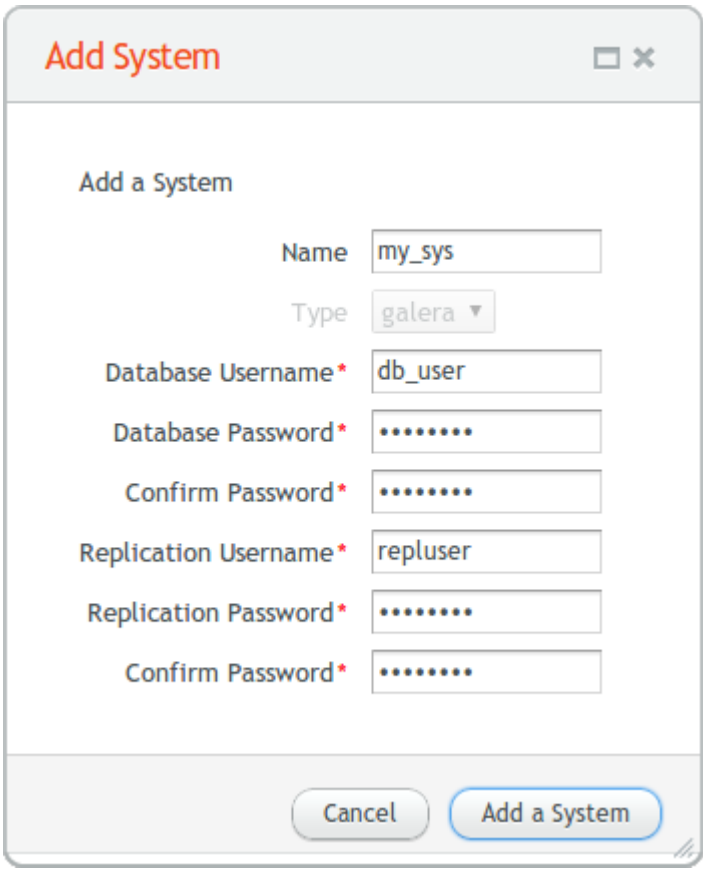
or

```
http://my.domain.example.com:8080/MariaDBManager
```

The first time you access the web user interface, you will be asked to provide a username and password for it to use to create the primary administrator account. **You should do this immediately after installation.** Until this administrator account is setup, the installation is vulnerable.

2.2.2 Adding a System

After creating the administrative user and using this account to log in to MariaDB Manager for the first time, you will see a message saying, No Systems Available. A system in this case is not a single server but is a set of replication servers, a cluster. Click on the button, Add System to add your first system. After clicking the button, you will see the following dialog box:



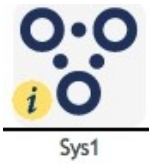
In the box labeled, *Name*, you enter the name of your system. We recommend something logical (and preferably short) that you will recognize and remember in relation to the set of servers that you will be

adding to the system. That said, this field is optional and if you do not enter a name, a generic one will be generated for you.

The next field is *Type*. At this time, the only choice available is *galera* so it is pre-selected and cannot be changed.

The remaining fields are all required. First, you must enter the username and password that you will be using for your databases. Then, put in the username and password that the nodes will use to communicate between each other for purposes of replicating between nodes. The database username and password will be used to monitor and control the cluster. If you use MariaDB Manager to install MariaDB Galera Cluster on your database nodes automatically, the administrative users will be created automatically using this information. If you plan to import existing nodes — ones that already host databases with data — you must put in the username and password of an existing administrative user common to all of your nodes. If you do not have a common administrative user with full rights set up on your existing nodes, you will need to do so before continuing.

When you have finished filling in the information about the system, click the *Add a System* button. You will soon see an icon similar to the one on the right appear in the interface. The name below the icon will be the name you set in the Add System dialog or the automatically generated name if you did not enter a name.



Remember: a system in this sense is not an individual server. It is a collection of servers operating together as a cluster. We refer to the individual servers which are part of the system/cluster as nodes in this document. Click the button, Done to complete this part of the process.

The next step is to add one or more nodes. A fully functional and available MariaDB Galera Cluster includes at least three nodes. The next section describes how to add nodes and to install and setup MariaDB Galera Cluster on those nodes.

2.3 Adding and Configuring Nodes

Now that you have created your first system, you are ready to specify and provision the nodes that will be part of that system. Nodes are servers on which MariaDB and Galera will be installed and which will contain application data. You can add existing servers which are already running MariaDB Galera Cluster, in order to simplify management of your already-configured clusters using MariaDB Manager. Alternatively, you can specify new and unprovisioned servers on which MariaDB Manager will install and configure all of the needed software. This process is fairly simple, and by letting MariaDB Manager handle all of the details, a large number of tedious and difficult manual steps can be eliminated. After you have used MariaDB Manager to fully provision a node, you will be able to start the node, monitor its activities, and control it through various administrative commands.

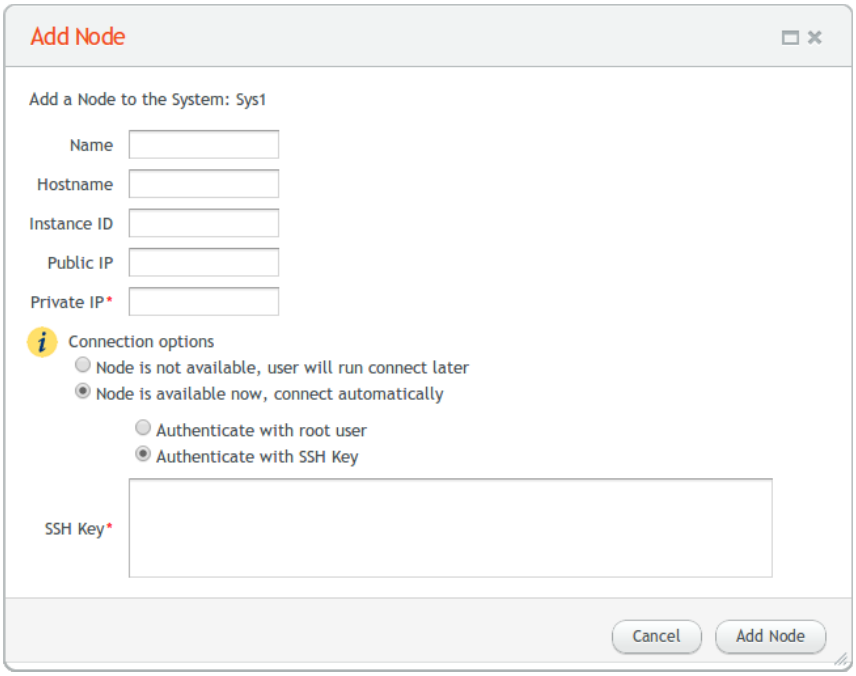
2.3.1 A note on provisioning

The process by which a node is added is called *provisioning* and is part of the initialization process of a new node, described in the *Initializing a Node* section later on in this document. There are two types of provisioning, *manual* and *automatic*. Automatic provisioning is recommended as this method

automatically handles all tasks related to installing and configuring the necessary software on a node. However, automatic provisioning cannot be used on a node that has MariaDB or MariaDB Galera Cluster already installed on it. If this is the case on any node which you intend to add to a system, you will have to use manual provisioning. With manual provisioning, you will install the necessary packages and make the appropriate configuration changes and additions. The necessary settings are detailed and discussed in the *Manual Provisioning* subsection of the *Initializing a Node* section of this document.

2.3.2 Adding a Node

To add a node, click on the Edit button located far to the right of the system icon within the web interface of MariaDB Manager. If this is the first node added to the system, there will be a message on the screen that says, *No Components Available, Press "Edit" to begin adding Components*. Click on the *Edit* button and then click on the *Add Node...* button to add a node to the system. You will see a dialog box like the following:



Apart from the *Connection options* section, the only required field in this dialog box is the *Private IP*. This is the address the management node will use to connect to the node you are creating. The *Name* field is not required. A name will be generated for you if you don't supply one here. This name is for all users who access the system, not for the server to use, so it does not have to match the server name or DNS entry. The *Public IP Address*, *Hostname*, and *Instance ID* fields may also be left blank.

If the node is up and running, select the *Node is available now, connect automatically* radio-button in the *Connection options* section. If not, select the *Node is not available, user will run connect later* radio button. Once the node is up and running you will be able to manually start the initial connection and setup processes.

Last, you have a choice to either to *Authenticate with SSH Key* or to *Authenticate with root user*. The following two sections discuss each of these methods, in turn.

2.3.3 Authenticating with an SSH key

The SSH key method is selected by default. To use this method, you paste in the private SSH key you wish to use, with the headers and footers, into the supplied box. For example:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAjkFz4YUjXlpQLD/WmwFOhOQcyVRqlMW/5XVXTx9f1nhJb3n07EtWTVrN
7f37NACOG5d2Yo+u1gUyv/YeaDZdiw7t7ZcyKDOd1xt1U1/aQSYPRd29ujEKsr5CmN0a1opk
...
9J6AtjRMgfZv+t2rrIW623HR8bpAMKuHk+hM5bPoRFuJ5w1H4qrPqeYdcvu2Pb0yXnQqXKkG
i4eb5xJnXxmfUW0/S3/0JLFT8Ryfu7rYz1dxPzIUX7MeG8aCTuY60g3bM3gafJrnCG91AA==
-----END RSA PRIVATE KEY-----
```

To keep this key safe it is not stored on the management node, or anywhere else in MariaDB Enterprise Cluster, after a node has been provisioned. However, the connection between your web browser and the management node is not encrypted. To encrypt this connection see the instructions in the section *Connecting securely to the MariaDB Manager*.

To provision using the SSH key method you must put a public SSH key (the one that corresponds to the private key which you pasted into the *Add Node* form) into the `/root/.ssh/authorized_keys` file on the node. It may also be necessary to modify the `PermitRootLogin` option in the `/etc/ssh/sshd_config` file on the node. It is likely this variable already exists in the config file. Using a text editor, look through this file to locate the option and insure that it is uncommented, and set to `yes` like so:

```
PermitRootLogin yes
```

If this option is set to `no` the `root` user will be unable to login via SSH at all. After modifying the `sshd_config` file, you must restart the `ssh` service on the node so that the change takes effect.

2.3.4 Authenticating with the root user

If you choose to use the *Authenticate with root user* method, you select that radio-button and enter the password of the system `root` user for that node in the text box. As with the SSH key method, you may need to edit the `/etc/ssh/sshd_config` file and set `PermitRootLogin` to `yes` as described in the *Authenticating with an SSH key* section previously. You may also need to set the `PasswordAuthentication` option to `yes` like so:

```
PasswordAuthentication yes
```

As with the SSH key method, the password is not stored on the management node, or anywhere else in MariaDB Enterprise Cluster, after a node has been provisioned. To encrypt the connection between your web browser and the management node see the instructions in the section *Connecting securely to the MariaDB Manager*.

You have the option, once a node is fully provisioned, to return any modified SSH settings to their previous values. When provisioning a node, MariaDB Manager on your management node handles the various provisioning tasks by logging in to the server with SSH as the root user. This SSH access is only needed during initial setup and provisioning. We plan to revisit this approach in a future version, to allow a different user account instead of root to be used in provisioning, which will remove the `PermitRootLogin` requirement.

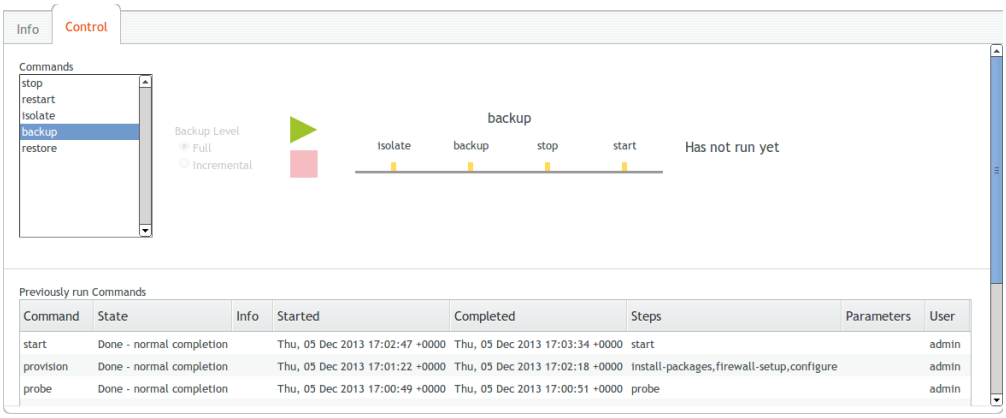
Unless you are manually provisioning nodes (not needed if using MariaDB Manager's automatic provisioning), or possibly changing some SSH settings, you do not need to log in to the database nodes during this process.

2.3.5 The node icon

When you are finished entering information about the new node, click the Add Node button. Then click the Done button. You will see an icon appear, similar to the one on the right. The name of the new node will be below the icon. If you click on the icon, the main panel will provide information on the node. However, since the node hasn't been completely provisioned yet, there will not be much to see just yet. To finish provisioning the node and to get it up and running, click on the Control tab.

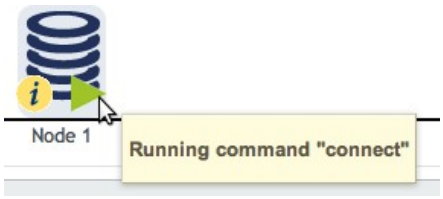
2.4 Control and Commands

Under the *Control* tab there is a box on the left with the heading of Commands (see the following screenshot). The contents of this box will change depending on what is available or possible based on the node and its current state. To execute a command, click on the name of the command and then on the green triangle-shaped *run* button. To stop a command that has started, click the red square *stop* button. When not available, the colors of these buttons are lighter. When a command is running, to the right of these buttons will be a progress indicator showing the progress of the steps involved. Some actions require more than one step. The overall command name will be positioned over the line, with the name of each step indicated with a marker and label on the line. The color of the marker will change based on the activity of the step: black for prior to a process starting; yellow for in progress; red for failed; and green for completed successfully. To the right of the progress indicator line is a general comment on the progress of the command intended to help further explain the process. The comment will update automatically as the command runs.



2.4.1 Initializing a Node

After you have created a new node, the only command available in the *Commands* box will be connect. If you selected the *Node is available now, connect automatically* option in the *Add Node* dialog box, the *connect* command will start automatically when the node is added. While the *connect* command is running, in addition to the progress status, a green triangle will appear on the node icon similar to the one on the right, indicating that a command is currently running on the node. This allows you to switch to another node and still see if the command you started on this node is still running. If you hover the mouse pointer over the node icon, you will see a pop-up tool-tip showing which command is running, like so:



When MariaDB Manager performs the *connect* command, it will check to see if the IP address and other information provided is correct, register information about the new node, install the agent scripts, and create users for itself on the node. When the *connect* command has completed MariaDB Manager will be able to interact with the node in order to run all subsequent commands and root SSH access will no longer be needed.

If MariaDB Manager is unable to connect to the node, the node icon will display a red triangle-shaped warning sign. If you hold the mouse pointer over this warning indicator, you will see a warning message similar to the one illustrated below, explaining the problem with details on what went wrong, for troubleshooting and repair.

Once MariaDB Manager has connected successfully to the new node, the *connect* command will no longer be shown and the only command available will be *probe*. Below is an illustration showing what this will look like in the *Control* tab. Click on *probe* and then click the run button to start the command. MariaDB Manager will then attempt to probe the node to ensure the node can support MariaDB Galera Cluster and other related software. It also checks for existing database installations, preventing accidental overwriting of application data.

2.4.1.1 Manual Provisioning

If the *probe* command discovers that MariaDB is already installed and running on the new node, MariaDB Manager will ask you if you are sure you want to make it part of the system. Care must be taken at this point, especially if you already have data on the node. Because of the way that Galera replication works, if this is not your first node in the cluster, then adding the node to a cluster will cause the existing data on the node to be overwritten by the data from the cluster. This is expected behavior — adding a node to an existing high-availability cluster will force that node to have a complete replica of the cluster data — the mechanism by which the cluster provides resilience to failure. Also, if MariaDB is already installed, MariaDB Manager will not make changes to the configuration or the installed software on the node in order to safeguard potentially valuable application data. In this event, required packages or configuration settings may not be in place. In order to add the node and allow it to be a part of the system you are configuring, you will need to manually configure the necessary software. This is the manual provisioning process.

If you are certain you want to manually add the node to the system, here are the steps you must take to successfully add it.

First, you must manually install all of the necessary packages. These are:

- `MariaDB-Galera-server`
- `MariaDB-client`
- `galera`
- `xtrabackup`
- `iproute`
- `net-tools`
- `grep`
- `findutils`
- `gawk`
- `nc`
- `coreutils`

You will also need a Galera configuration file. This file must be placed under the `/etc/my.cnf.d/` directory. The file can be named anything you like, ending with a `.conf` extension. Some name suggestions: `galera.conf` or `mariadb-galera.conf` may help from a maintenance perspective. An example configuration file is in *Appendix B: Example Galera Configuration*.

With the packages installed, you must then create a database user and a replication user on the node. These users must match the values entered when the system was created in the MariaDB Manager WebUI. For example:

```
CREATE USER 'dbuser'@'127.0.0.1' IDENTIFIED BY 'password';
GRANT ALL ON *.* TO 'dbuser'@'127.0.0.1';
CREATE USER 'repluser'@'127.0.0.1' IDENTIFIED BY 'password';
GRANT ALL ON *.* TO 'repluser'@'127.0.0.1';
```

Next, you will need to install Xtrabackup. Instructions for installing this package are available at: http://www.percona.com/doc/percona-xtrabackup/2.1/installation/yum_repo.html

Xtrabackup requires the following setting to be in the `[mysqld]` section of the `/etc/my.cnf` file on each node:

```
datadir=/var/lib/mysql
```

Also, please refer to the *Ports* subsection of the *Installing MariaDB Manager* section of this document for information on the ports and services that must be open or enabled.

Lastly, the `selinux` policy on each node must be set to *permissive* or *disabled*.

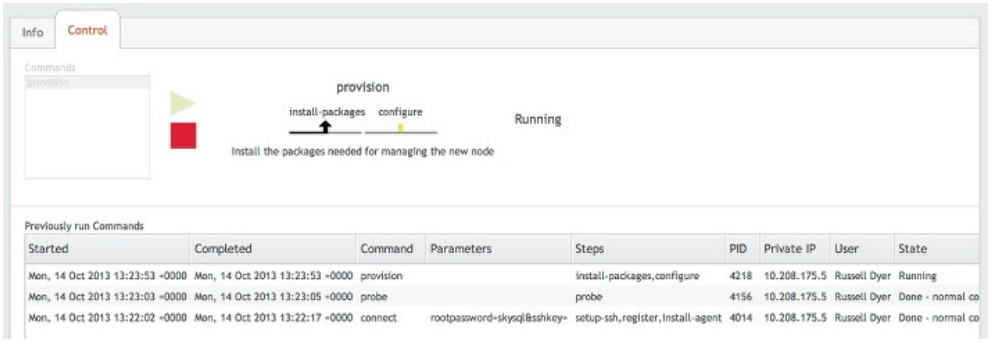
2.4.1.2 Automatic Provisioning

If the *probe* command determines that the new node does not have MariaDB or related software installed on it, MariaDB Manager will install the needed software on the node and make all necessary configuration changes and additions. There is no need to download or manually install and configure the software. This process is known as *automatic provisioning*.

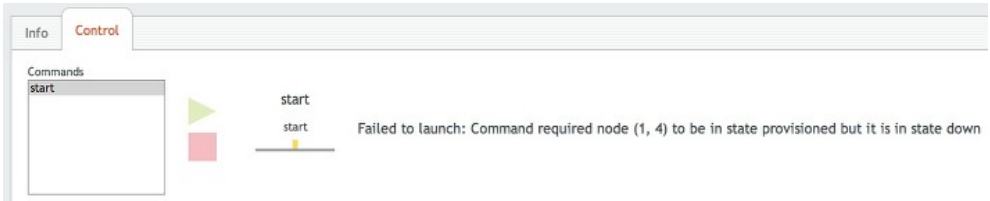
When MariaDB Manager is installed on the management node, it locally caches a complete set of RPM packages for the software it will provision onto the nodes. With each node installation, it uses these same RPM packages so that all of the nodes are using the same software versions and are therefore compatible with each other. Caching the RPMs locally also improves provisioning speed, as packages are accessed over the faster local-area network, rather than over the Internet.

2.4.2 Starting a Node

Once you have successfully provisioned a node, the command box will show the *start* command. It will be the only command available. Click on that command, then click the *run* button to start MariaDB Galera Cluster on the node.



If for some reason the node fails to start, the icon for the node will change to a red version like the one illustrated on the right. Additionally, you will see a message on the screen which says, *Failed to Launch*, along with other information to indicate how far into the start process the node progressed before the failure.

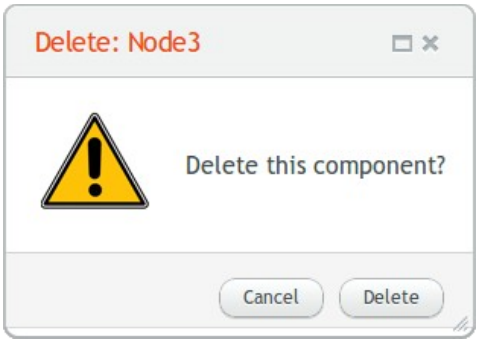


There are many potential causes that could prevent a node from starting. If a node fails to start, click on the red *stop* button to stop MariaDB Manager from continuing to try. Then select the command *start* and click the *run* button again. If this doesn't solve the problem, check the system logs on the node for clues as to the cause. If you entered something incorrectly when creating the node, you should delete it from MariaDB Manager and then recreate it.

2.4.3 Deleting a Node

To delete a node, click on the *Edit* button, located to the right of the section where the node icons are shown. The node icon will change to look like the one illustrated on the right. There are two additional graphics superimposed on the icon: one which looks like a pencil, which is to edit the node's settings; and the second with an X in a red circle, which is used to delete the node.

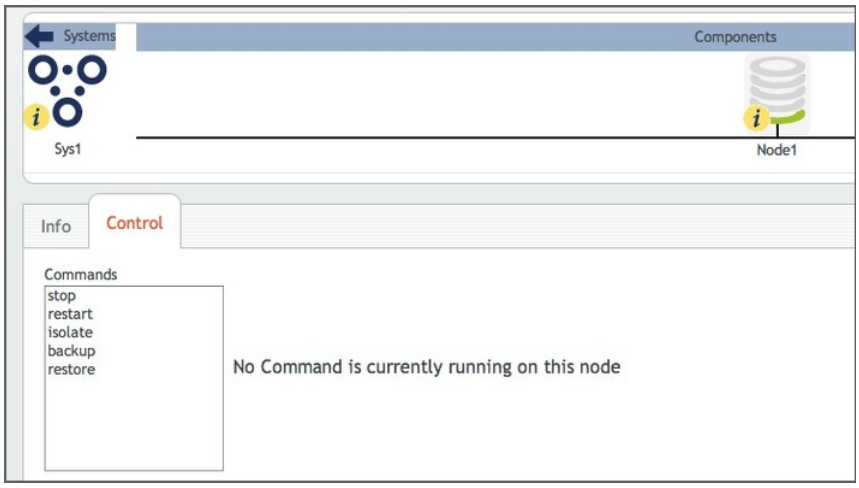
If you click the red X to delete the node, MariaDB Manager will display a confirmation dialog. Deleting a node removes it from the system and it will no longer receive updates through replication. However, after deletion, the server will still be provisioned with the MariaDB Galera Cluster software along with any data in the node's database prior to being deleted.



If you entered the IP address of a node incorrectly when you created it and want to delete and then recreate the node, you will need to edit the Galera configuration file located under `/etc/my.cnf.d/` on the node you are re-provisioning to fix the IP address.

2.4.4 After a Successful Start

If the node starts without problems, then the node's icon will look a bit green like the icon in the top right corner of the following screenshot. Several commands to administer the new cluster node will now also be available in the *Commands* box. These commands include *stop*, *restart*, *isolate*, *backup*, and *restore*. The *stop* and *restart* commands are self explanatory. The *isolate*, *backup*, and *restore* commands will be described in the next section. All of these commands require the node to be running. If you *stop* the node, only the *start* command will be available.

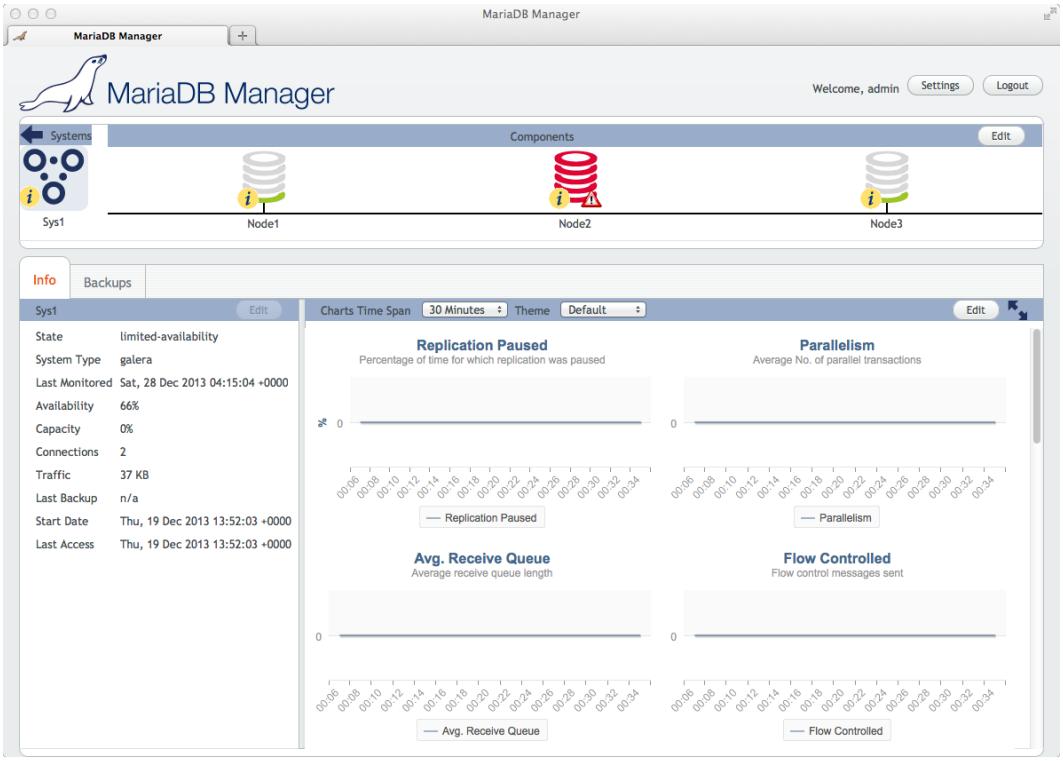


3 Monitoring Systems

The main screen is divided in two main areas: the Navigation panel, where the system composition can be seen and modified by adding systems and nodes, and the Dashboard just below it.

3.1 Using the MariaDB Manager Dashboard

The Dashboard area is a tabbed panel where monitoring and control of system resources are performed. The tabs that are available in this panel are dependent on what is selected in the Navigation panel and the information displayed in these tabs is directly relevant to the resource (System or Node) selected in the Navigation panel.



3.1.1 Info tab

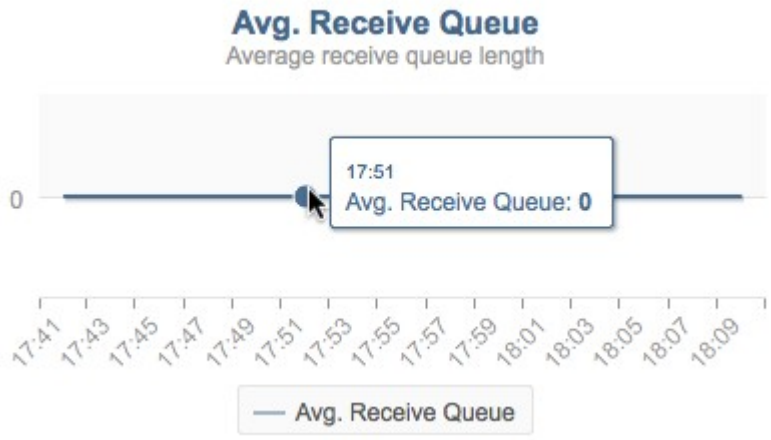
The Info tab is initially displayed when the user logs into the application. It displays information related to the selected system or node and is refreshed periodically with the current information.

This panel is split vertically and the drag bar allows to change the area assigned to each side.

A summary of some of the relevant info for the selected component is displayed in the narrower left side. The header of this section contains the name of the selected component as a reminder of what is selected in the navigation panel.

The right side of this panel is used to display the user's selection of charts, each containing the data collected by one or more monitors. Upon the initial user login, a series of charts will be created from the list of available system-wide monitors at the time, one for each SQL monitor. The user can subsequently elect to edit, add and remove charts from the series, rearrange their display order and group multiple monitors in the same chart.

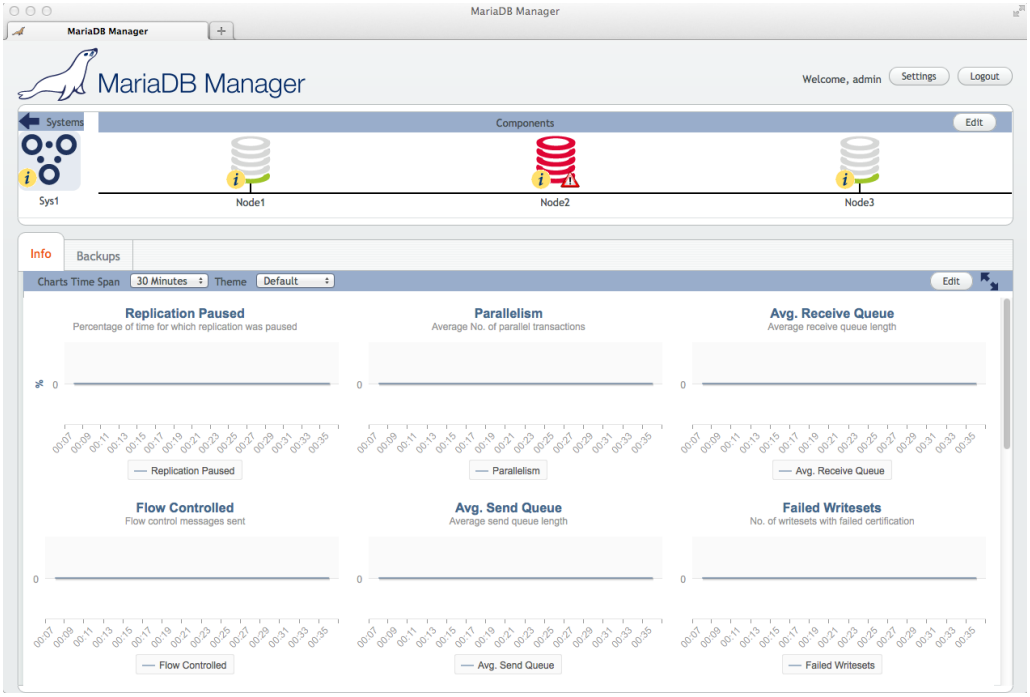
Hovering the mouse pointer over a chart allows the user to discover the exact value represented by any given point of a line chart, or the min and max set of values filled in an area chart.



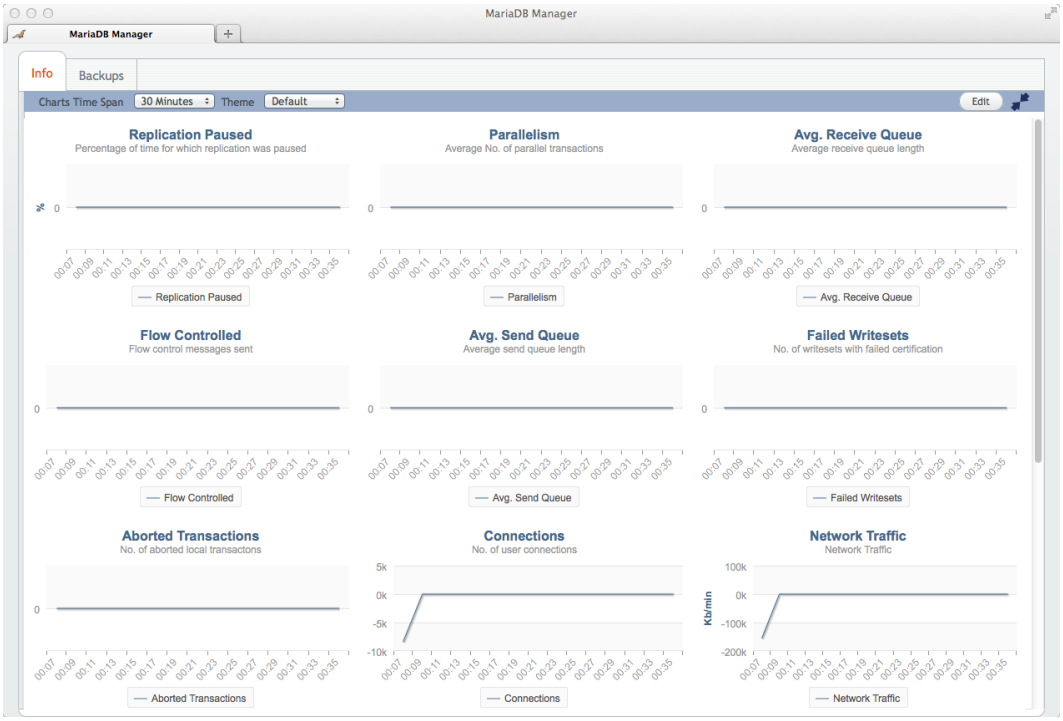
The user can change the timespan used for all the displayed charts by selecting it from the drop down in the layout's header. The rendering theme can be changed here too.

3.2 The Full-screen Dashboard

In order to use all of the horizontal space for displaying charts, the vertical tab in the Info panel can be dragged to the left-hand side.



A click on the expand button  in the right corner of the panel's header, allows to expand the panel to the full screen. Another click  will reduce it back to the previous size.

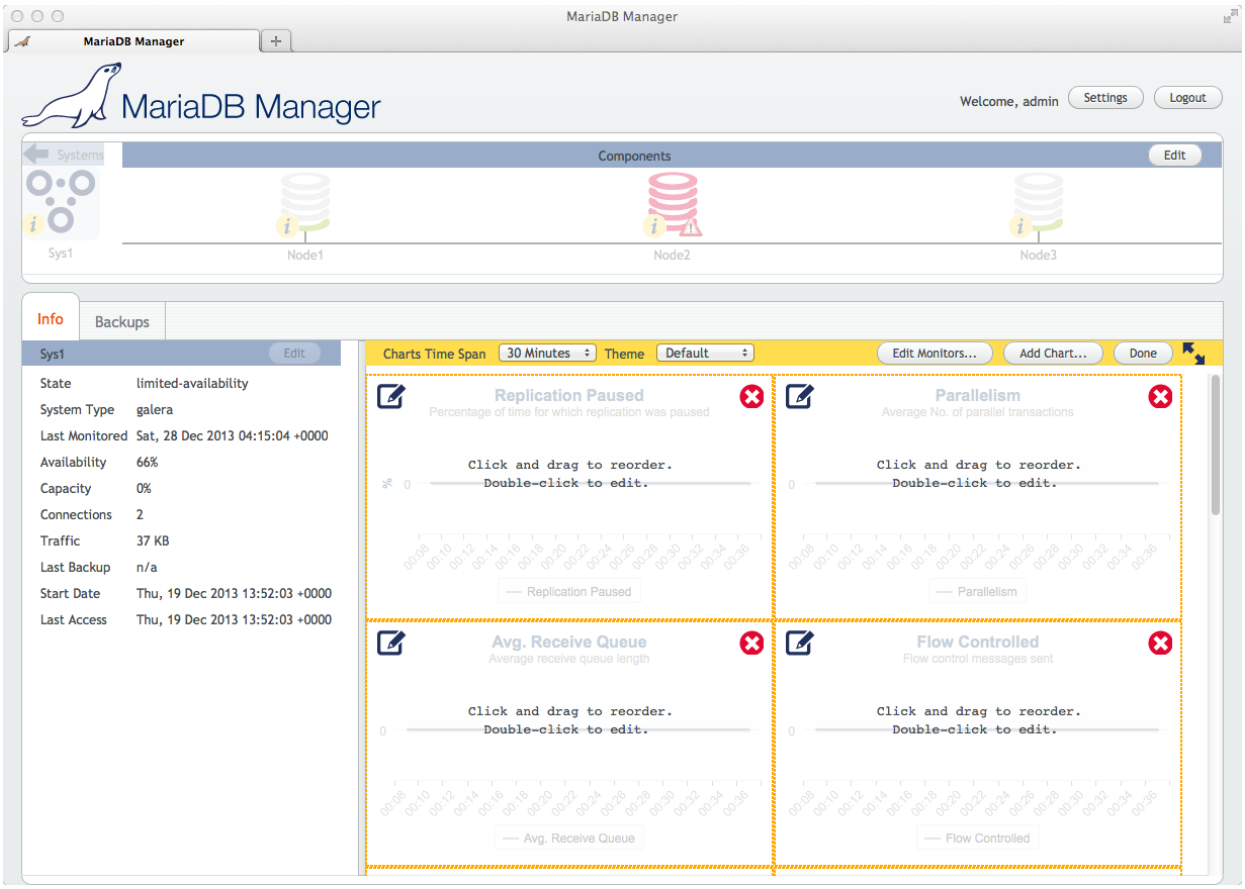


3.3 Choosing Monitors

The application initially displays a series of predefined charts based on a series of predefined monitors. To change this click on the Edit button in the right corner of the panel's header.

In *Edit* mode it is possible to change the series of charts displayed:

- to delete an existing chart, click on the red x icon in the right corner of the chart (this will only remove the chart from the display, not delete the monitor(s) displayed therein).
- to edit it, click on the blue icon in the left corner or double-click on the chart.
- to add a new chart, click on the Add Chart button in the panel's header. The new chart will be appended at the end of the series.
- to rearrange the display position of a chart, click on the chart, drag it and drop it in the new position.

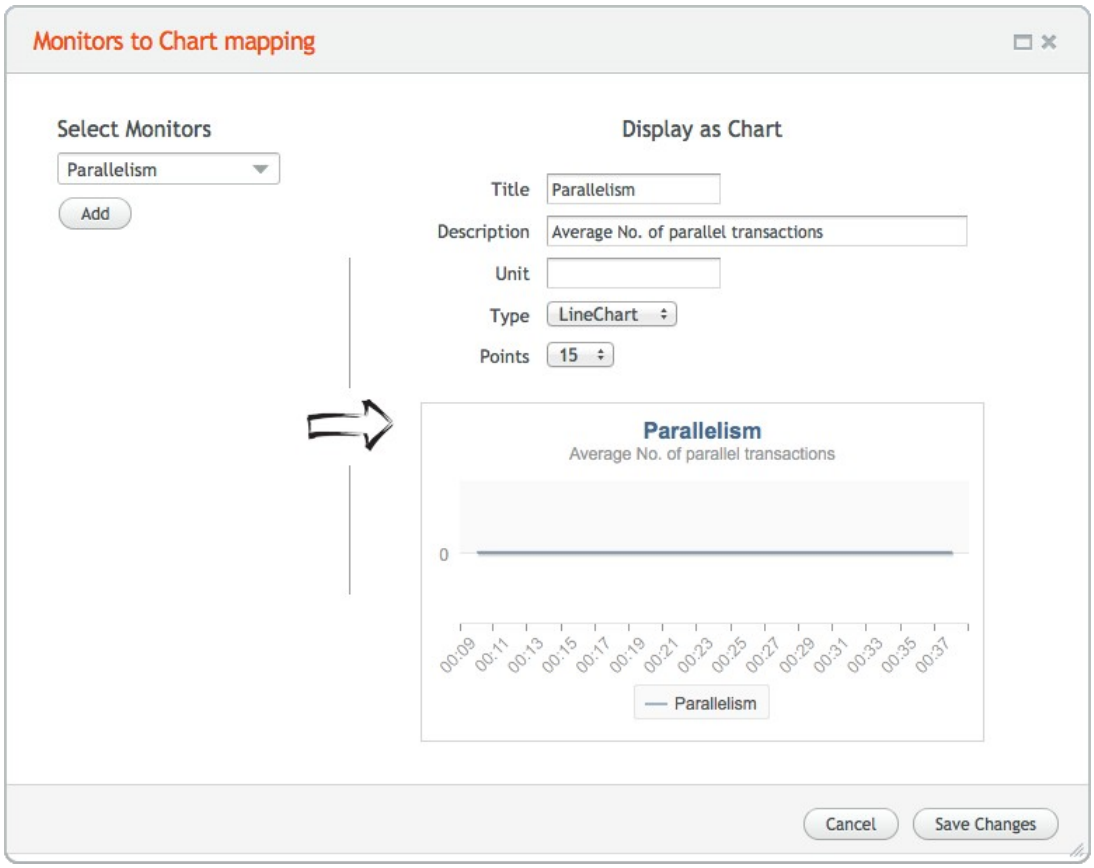


A complete list of available monitors is at: https://mariadb.com/kb/en/mariadb-manager-monitors/

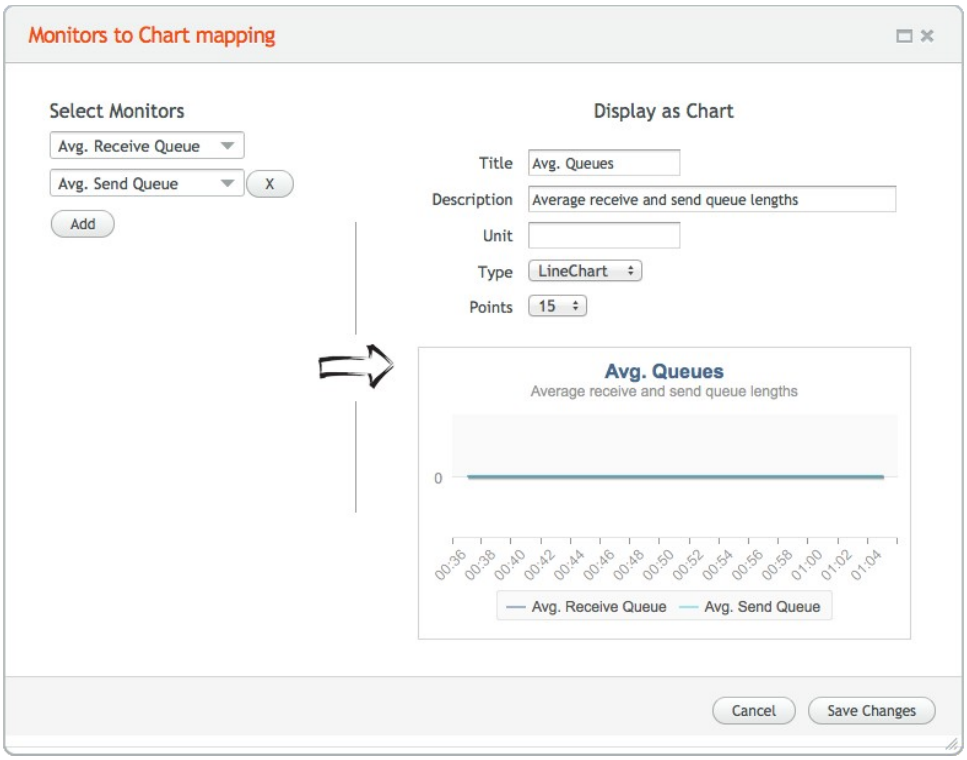
3.4 Previewing Monitors

3.4.1 Monitors to Chart Mapping dialog

Editing or adding a new chart will bring up a dialog where the user can select one or more monitors to be displayed in the chart.



Clicking on the *Add* button in the Monitors column (on the left hand-side) will add another monitor entry to the chart. The monitor can then be chosen from the dropdown.



The chart's *Title* and *Description* can be customized on the right hand-side.

The *Unit* field will be used as the caption for the vertical axis of the chart and will typically represent the unit of measurement used for the data returned by the monitor.

The chart type can be changed using the dropdown (*LineChart*, *AreaChart*, etc...).

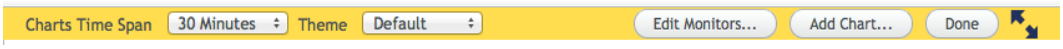
LineChart will display the monitor data by retrieving a computed average value for a discrete period of time and plotting a line through the series of values.

AreaChart will display the data by retrieving the minimum and maximum values observed in a discrete period of time and fill the graphic's area between the two.

The chart Points will determine how many data points will be used to render the chart for the given interval of time, by requesting as many from the API.

3.5 Installing New Monitors in the Dashboard

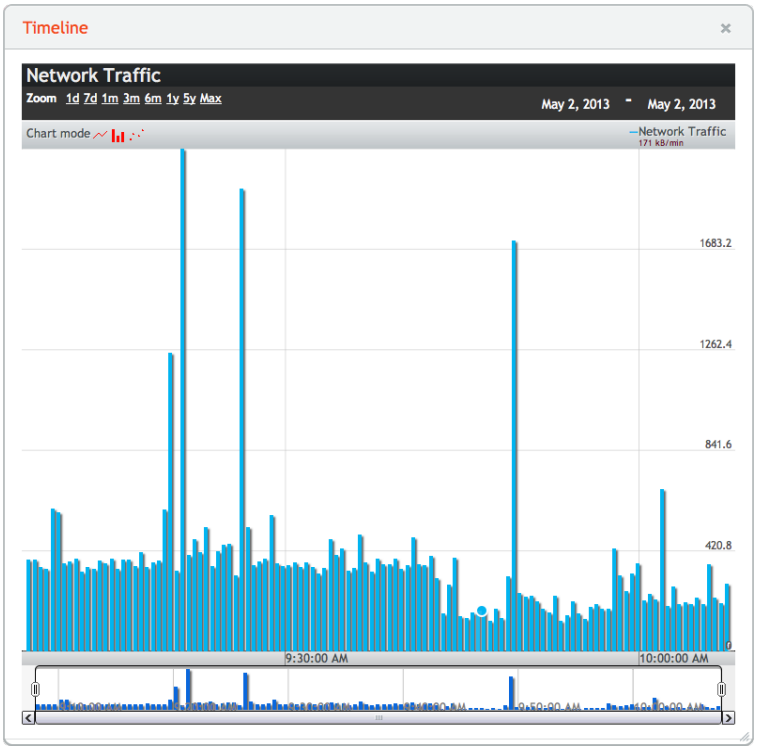
In the Edit mode it is also possible to access the list of existing monitors in order to view, edit, delete and add them to the system, by clicking on the Edit Monitors button in the panel's header.



This will access the Monitor tab of the Settings dialog (see detailed description in the Configuration chapter). After creating a new Monitor, it can be added to the dashboard as part of an existing chart or in a new chart by using the Add Chart button. To terminate editing operations on Charts, click on the Done button in the right corner of the panel's header.

3.6 How to use the Timeline

When viewing the Dashboard in its normal mode (not the editing mode), double-clicking on a Chart is going to bring up the Timeline window, which is used to inspect the collected monitor data in the greatest detail possible for the whole time span that the data has been collected for. Unlike charts which display data points computed from the stored observations collected by the SkySQL Monitor app, the Timeline pulls the raw data that was collected via the API and displays it with no or a minimal amount of alterations.



The time window can be changed by using the Zoom links at the top of the window or by moving (click and move) either end of the scrollable area at the bottom of the screen. This area can also be dragged around in its entirety in order to move the time window around.

4 MariaDB Manager Monitor Configuration

The MariaDB Manager Monitor is designed to start automatically on system startup and after installation of the MariaDB Manager package. The default settings of the MariaDB Manager Monitor package allow full interoperability with the other components of the MariaDB Manager package.

The Monitor settings can be found in the following files:

Monitor settings:

- `/etc/init.d/mariadb-enterprise-monitor`

Monitor API ID/Key pairs:

- `/usr/local/skysql/config/components.ini`
- `/usr/local/skysql/config/manager.json`

4.1 Configuring the init script

The primary monitor settings are found in the init script located at:

- `/etc/init.d/mariadb-enterprise-monitor`

The available options in this script can be summarized as follows:

Variable name	Meaning / effect	Default value
`vMonitorDir`	The directory where the Monitor is installed.	`/usr/local/skysql/monitor`
`vMonitorJar`	The filename of the Monitor main file, stripped of the path.	`ClusterMonitor.jar`
`vShareDir`	The skysql shared directory, which contains the MariaDB-Manager Java Library for proper functioning.	`/usr/local/skysql/share`
`vShareFile1`	The MariaDB-Manager Java Library file name, stripped of the path.	`skysql-java-shared.jar`
`vJavaClasspath`	The Java Classpath. It should contain at least the full pathed Monitor filename, and the MariaDB-Manager Java Library filename. On Windows systems, the colons (`:`) should be replaced by semicolons (`;`).	`"${vMonitorDir}/${vMonitorJar}:${vShareDir}/*"`
`vJavaSettings`	The Java Properties to be passed to the Monitor application. Please do not change the default language settings because they are necessary for proper functioning and interoperability with the API. Further recognized Properties are: 1. `SKYSQL_API_HOST` 2. `SKYSQL_API_KEY` 3. `SKYSQL_API_KEYID` They are described in a related document.	`"-Duser.language=en -Duser.country=GB"`
`verbose`	When this variable is empty (`""`) or not set, the Monitor will not show many lines in the log. If this it is set as `"-v"`, the log verbosity is increased.	`"-v"`

4.2 Configuring the Monitor API ID/Key pair

MariaDB Manager components make use of a securing system to avoid man-in-middle attacks. The system is based on a set of keys, each labelled by an ID. The keys may be found in the following files:

`/usr/local/skysql/config/components.ini`

This is a plain text file whose lines represent API ID/Key pairs. If you want to change or add an API key, we recommend that you choose to edit this file.

`/usr/local/skysql/config/manager.json`

This file provides configuration information according to the JSON format. Only the MariaDB-Manager WebUI component uses it, despite the Monitor also supports this file, so we strongly discourage using this file.

These files are discussed in detail in the following two sections.

4.2.1 The components.ini file

In the components.ini file a typical line is:

```
5 = "tgdehuj38gu58tikv893nmc71jf09itnc"
```

The number on the left is the `ID` of the key. The string enclosed in the double quotes is the key itself. If you want the key to be recognised by the API, you should write exactly the same line in the `/etc/skysqlmgr/api.ini` file on your API machine, in the section labelled `[apikeys]`. You will find this section at the end of the `api.ini` file itself.

If you install the MariaDB Manager package from scratch, the necessary set of API keys will be created automatically and placed in the `api.ini` file, too. The keys are generated during the package installation and are random 32 character long strings.

Each component of the MariaDB-Manager package is associated with a different `ID`. The association cannot be configured at this time. There are plans to allow modifying the ID/component map in a future release. The ID to component mapping is as follows:

Component	API Key ID
WebUI	1
Agent	2
Monitor	3
Agent on the Data nodes	4

4.2.2 The manager.json file

This file is formatted according to the JSON standard. It is used by the WebUI, and it is necessary.

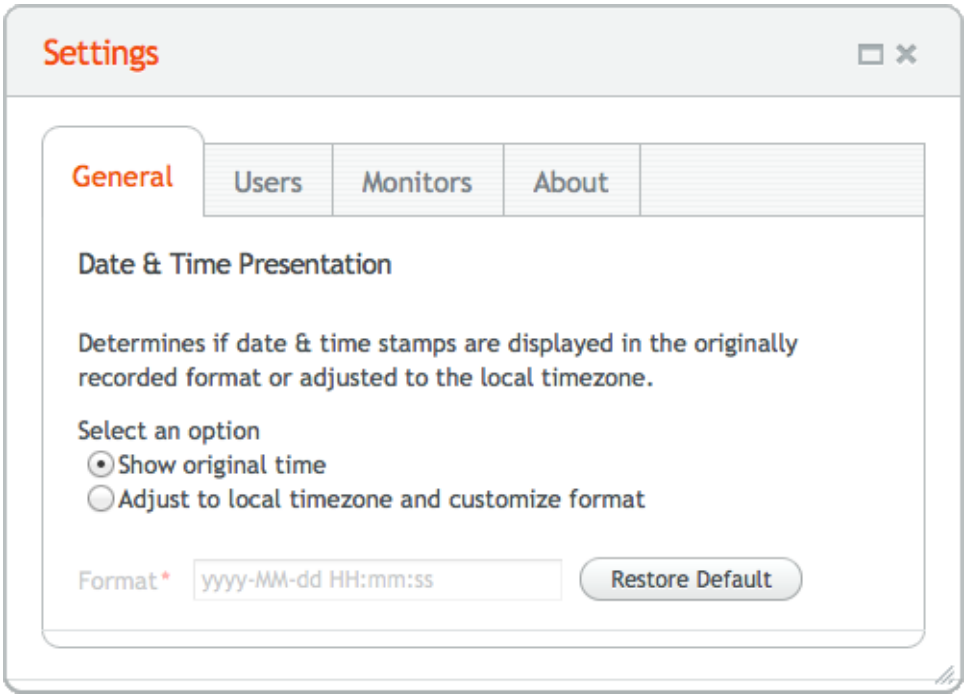
The WebUI key is defined in this file, so to modify the WebUI key, simply modify the corresponding entry in the JSON and also modify the `api.ini` file accordingly, following the instructions given previously.

5 Configuration & the Settings Dialog

The system and the Web client can be configured by clicking on the Settings button in the top-right of the application, which will bring up the Settings Dialog.

5.1 The General tab

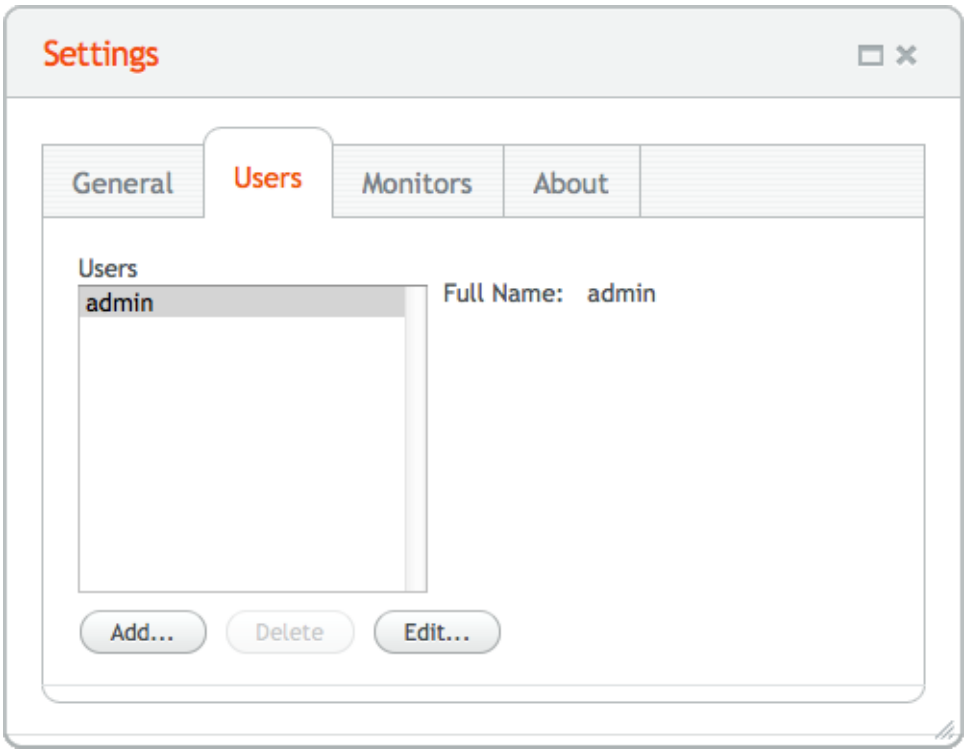
The General tab allows a user to customize the presentation of the date and time throughout the application.



By default it will display these as returned by the API, but users can choose an alternate format by using a pattern crafted after Java's `SimpleDateFormat`, as documented here: http://docs.oracle.com/javase/6/docs/api/java/text/SimpleDateFormat.html

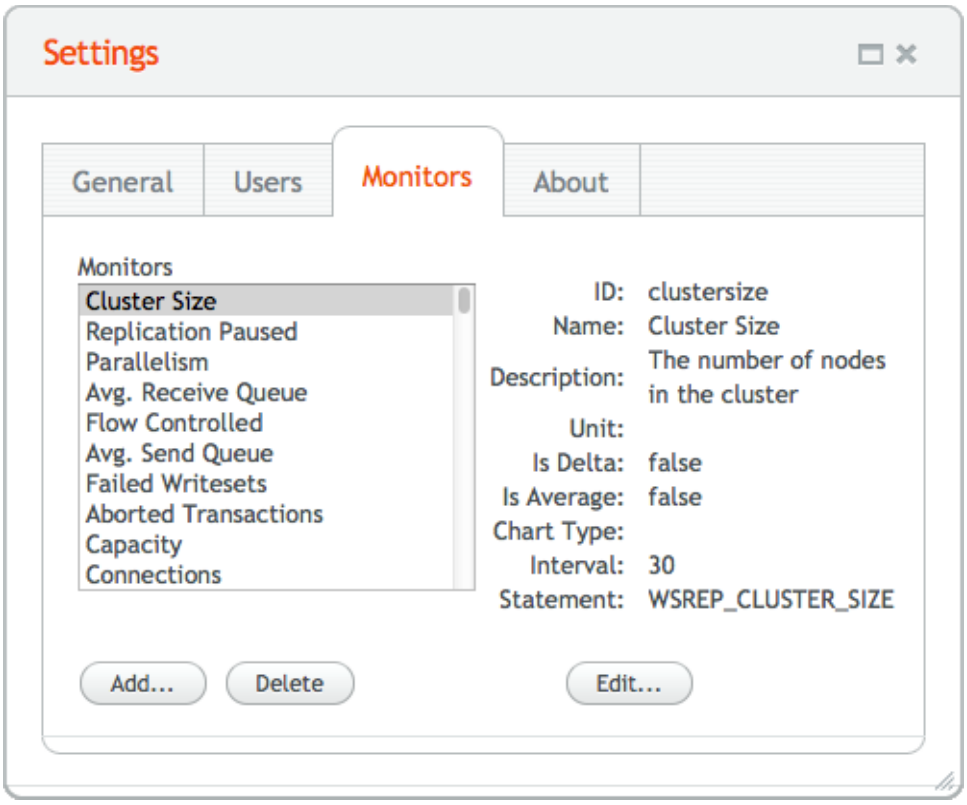
5.2 The Users tab

The Users tab allows the user to change the full name and password of an existing user, add a new user or delete an existing one.



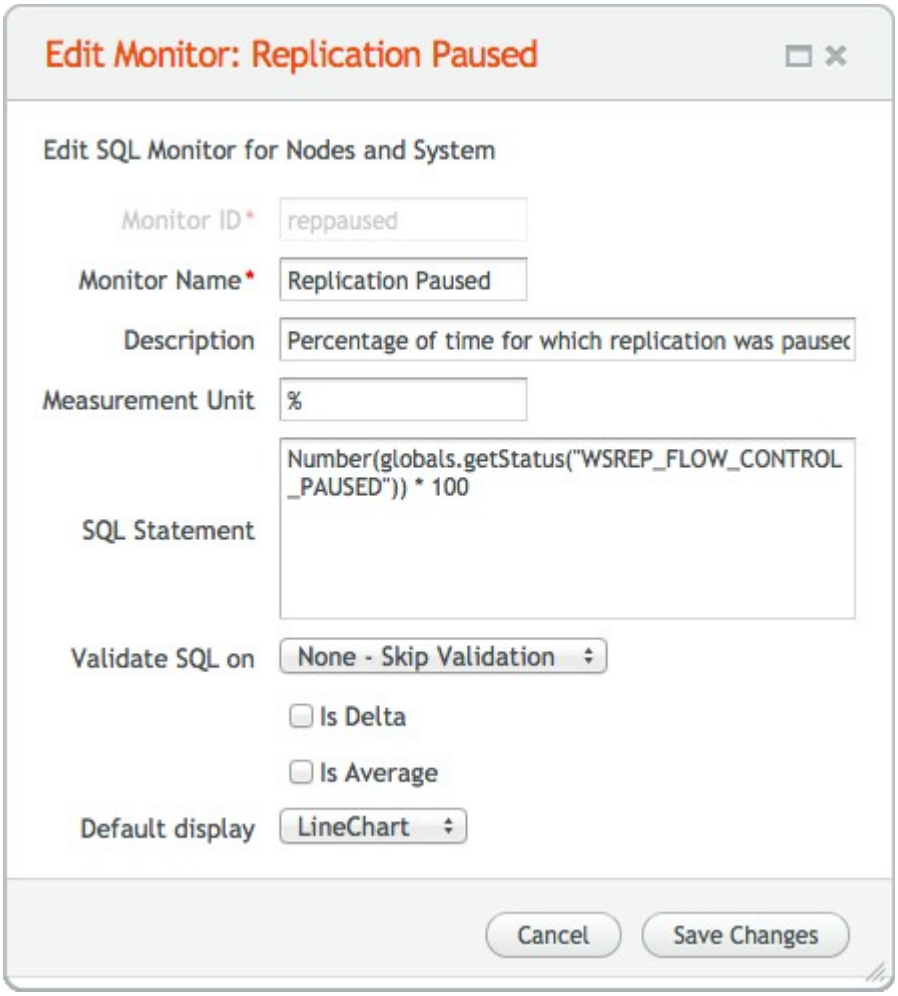
5.3 The Monitors tab

The *Monitors* tab allows the user to edit, add, or delete monitors associated with a particular system type. The currently selected or opened system will be used to determine which set is presented. See Monitoring Systems for a detailed discussion on how they operate.



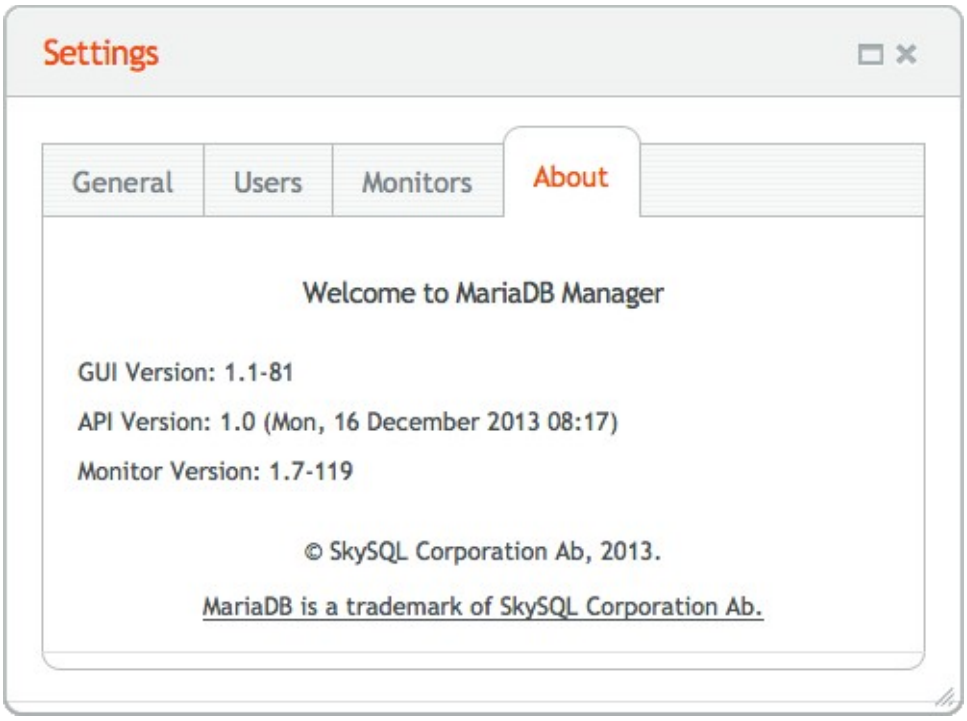
When adding a new monitor or editing an existing monitor, the properties are as follows:

- **Name** — for display purposes
- **Description** — for display purposes
- **Measurement Unit** — indicates the unit in which values are captured by the monitor and it is displayed as a label for the Y axis of a chart that displays the monitor values.
- **SQL Statement** — the SQL statement that is run against the database by the SkySQL Monitor app.
- **Validate SQL dropdown** — the node, if any, on which to validate the SQL statement prior to saving it. This helps ensure that the monitor will actually be able to run and collect the data that it is meant to.
- **Is Delta** — used to determine whether the monitor will collect absolute values or the change (delta) from the previous reading
- **Is Average** — used to determine if the values collected by this monitor for multiple nodes and added up to represent the cluster-level value are going to be left as the total sum or divided by the number of nodes in the cluster.
- **Default display** — determines which chart type is going to be pre-selected when choosing to display a monitor.



5.4 The About tab

The About tab provides info about the installed system.

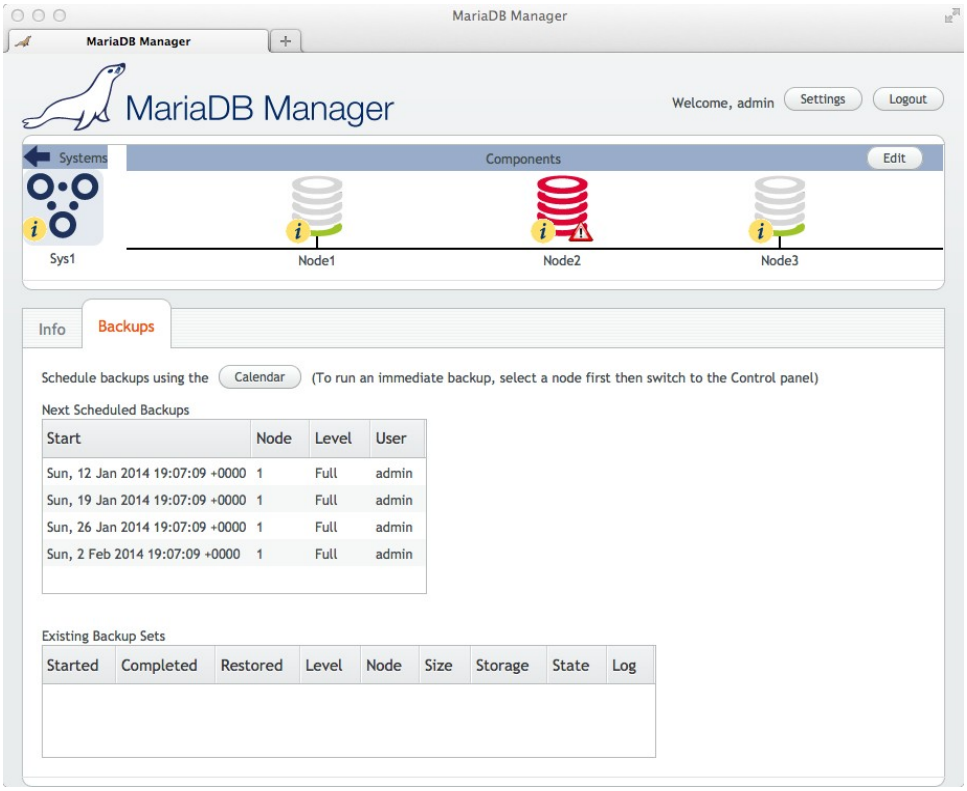


6 Backups

The backups tab is available when the System is selected in the navigation window. Its purpose is to manage and monitor the execution of a scheduled backup strategy.

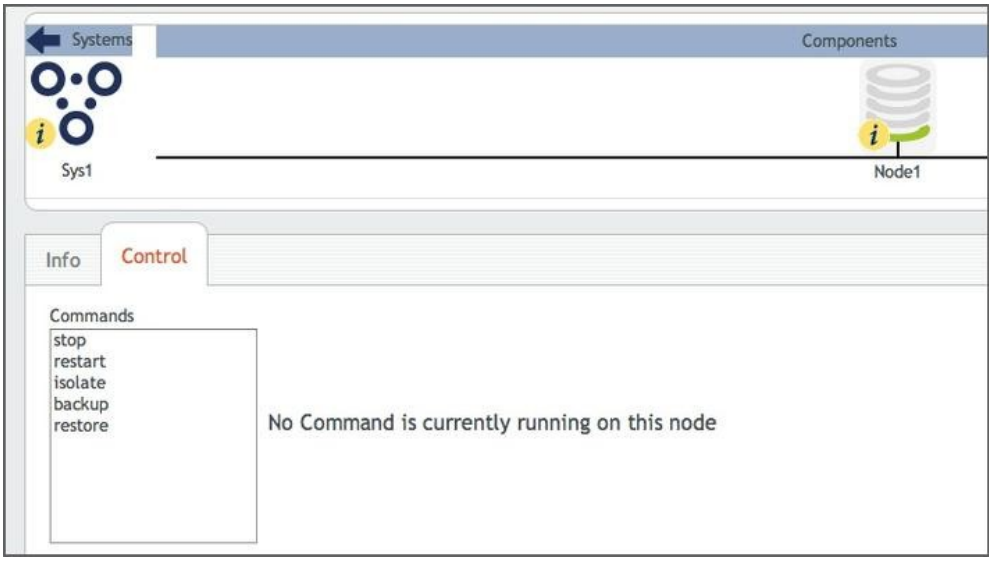
The panel is split horizontally:

- The top part is used to manage the scheduled backup strategy.
- The bottom part of the panel displays a list of available backup sets alongside the relevant information.



6.1 Isolating and Backing up a Node

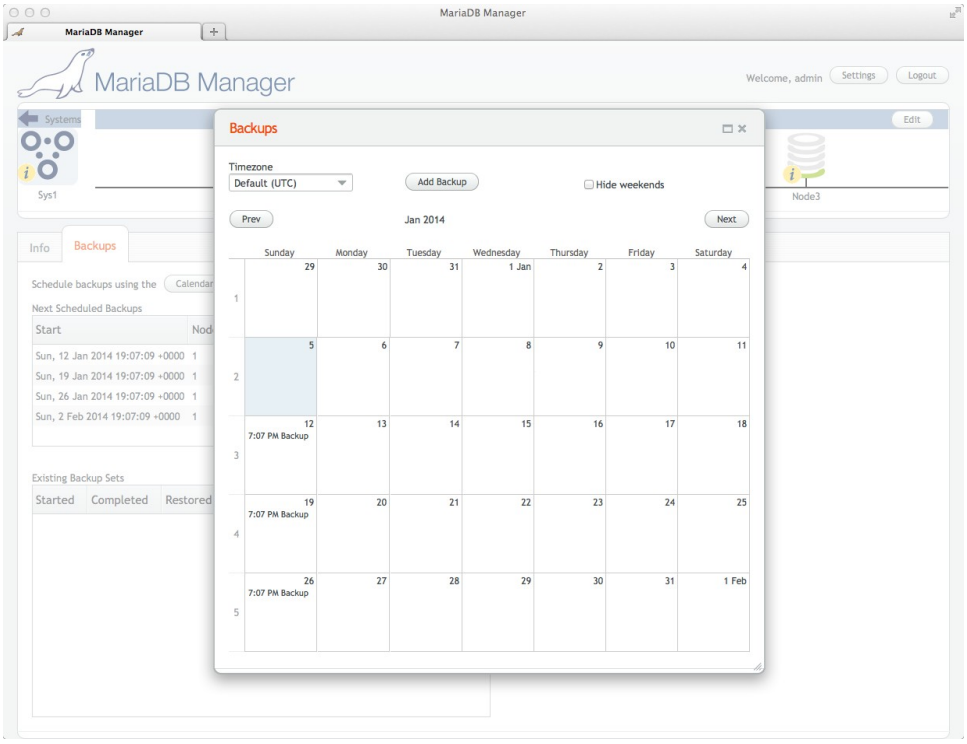
Once there are several nodes configured and running with data on them, if you want to make a backup of a cluster's replicated data, it is best first to isolate one of the nodes in the cluster, to stop replication traffic and ensure that the database remains consistent, and to prevent backup activities from affecting the other cluster nodes. First, run the isolate command and then, once the node is isolated, click on backup to make a backup of the databases on the node. If you run the backup command without running the isolate command first, MariaDB Manager will automatically run the isolate command before running the backup command.



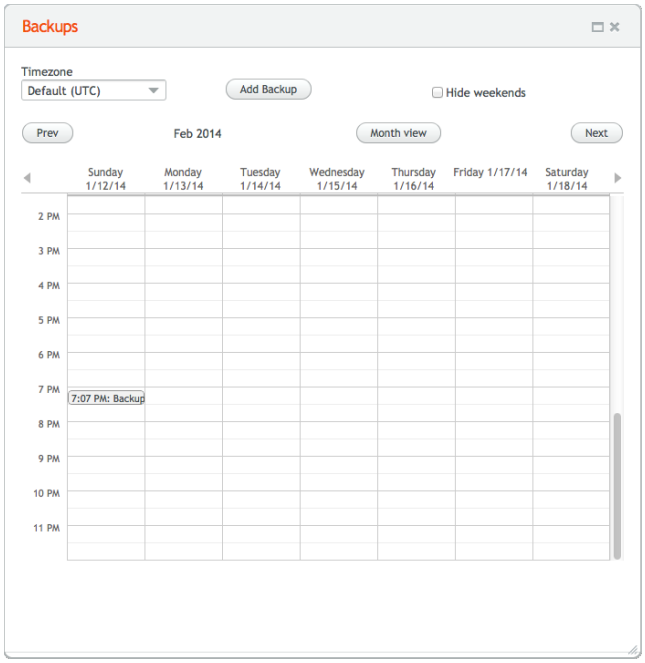
When the backup is finished, click on rejoin to have the isolated server rejoin the mix of nodes for the system. The other nodes in the cluster will then provide snapshots to the node rejoining the system to synchronize it and bring it up-to-date with the other active nodes — not SQL statements like normal MySQL replication would provide.

6.2 Scheduling Backups

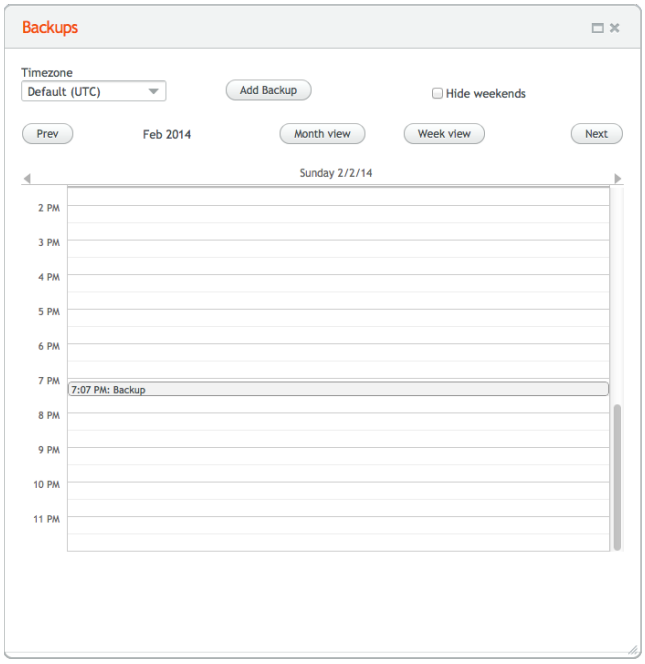
Clicking on the Calendar button brings up the corresponding window where scheduled backups can be created, viewed, edited and deleted.



The calendar can viewed in Month (default), Week (click on the *week number* on the left hand-side) or day (click on the *day number*) modes. The *Week* mode looks like the following screenshot:

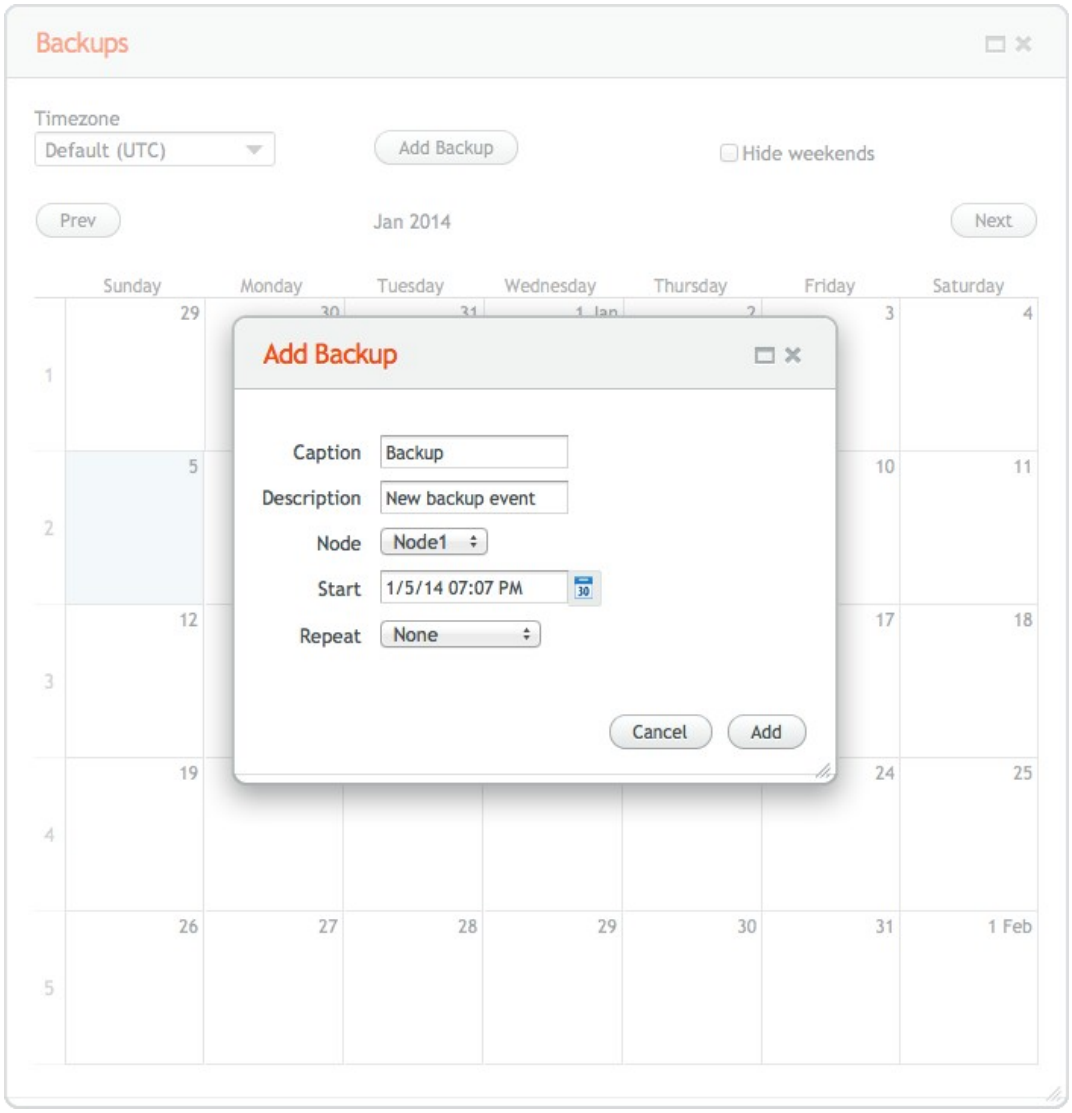


The Day mode looks like the following screenshot:

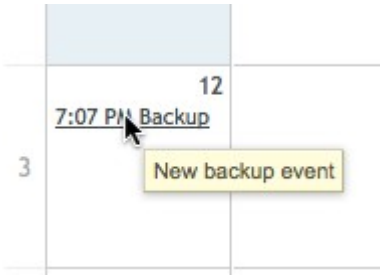


The timezone used to view the scheduled events can be changed by the user for convenience.

To create a new scheduled backup, either click on the Add Backup button or click into the calendar on desired day/time and the corresponding dialog will appear.



The *Caption* is what is displayed on the Calendar; the *Description* appears when the mouse hovers over a calendar entry.



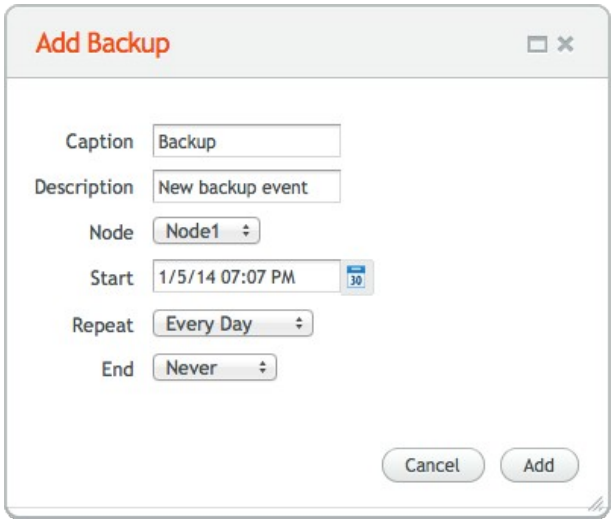
The *Node* selection will determine which one is used at the time the backup is performed.

Start will determine when the individual backup is performed or when a repeating series of backups starts, depending on the *Repeat* selection.

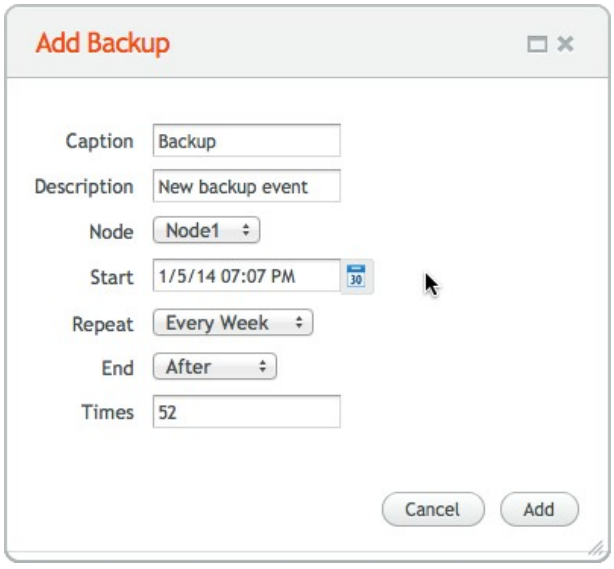
Schedules can be repeated on a daily, weekly, monthly and yearly basis.

When choosing a repeating schedule, the *End* dropdown will determine when to end the series: *Never*, *After* a certain number of occurrences, or *On* a certain *Date*.

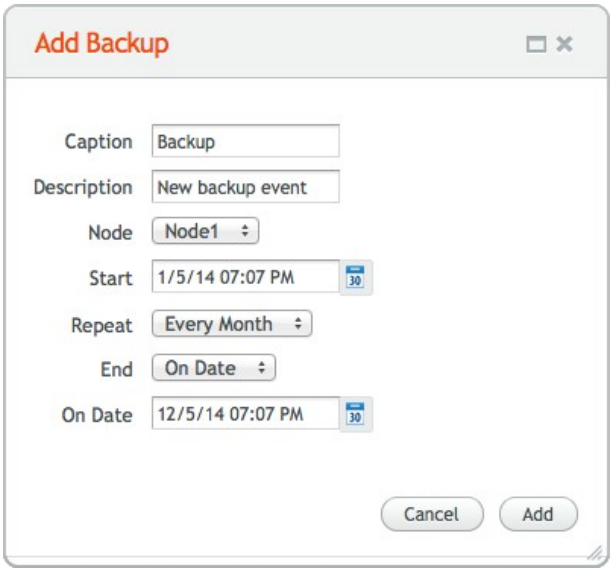
With *Never* selected, the dialog looks like the following:



With *After* selected the dialog looks like the following:



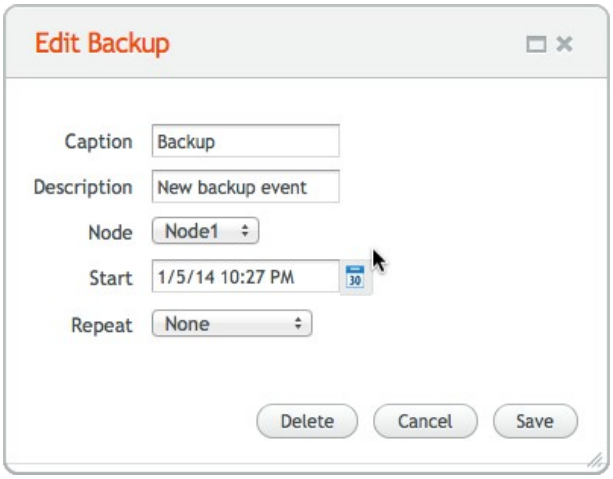
And with *On Date* selected the dialog looks like the following:



Once added, the single or repeating schedule will be visible in the Calendar window.

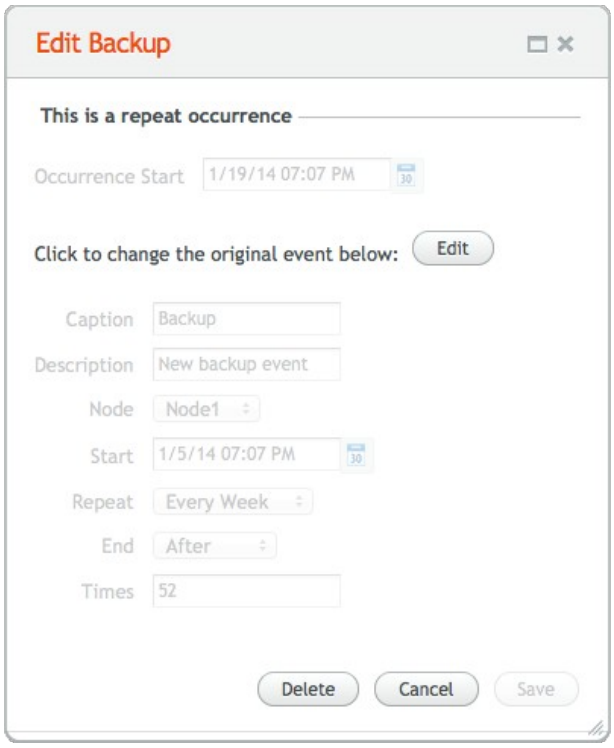
6.2.1 Single occurrence

When clicking on an existing schedule entry in the Calendar, the same dialog will be brought up to view or edit a single schedule, and clicking on the Delete button will remove that schedule entry.

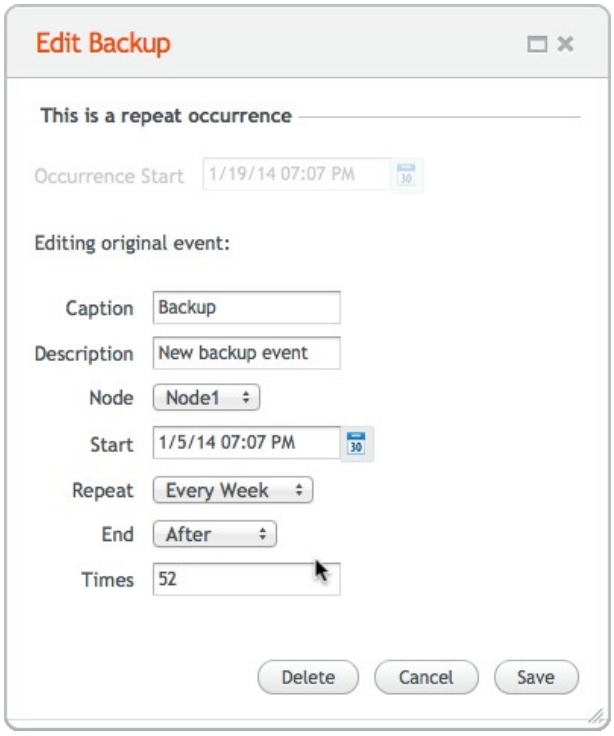


6.2.2 Repeat occurrence

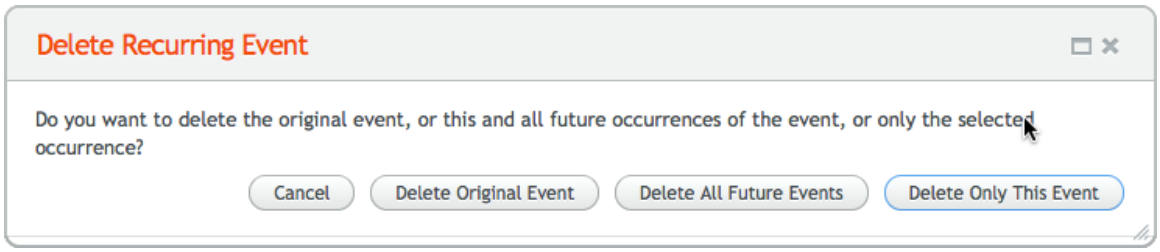
A modified version of the dialog will be brought up to view the particular occurrence of a repeating series. The latter will have an *Edit* button in order to allow the user to modify the whole series. Before clicking the button, the dialog will not allow you to change the fields.



After clicking on the *Edit* button, the dialog will allow the fields to be edited.



While editing, clicking on *Delete* will bring up a multiple choice dialog to allow removal of just that particular occurrence, or that event plus all future occurences, or the whole repeating series.



6.3 Restoring from a Backup

To restore from a backup to a MariaDB Galera Cluster, you must perform the following steps in the correct order, or you will not successfully restore your data:

4. Isolate all of the nodes in the system.
5. Execute the restore command to restore the data onto a node of your choosing.
6. Run the rejoin command on the node you just restored.
7. Run the rejoin command on the other nodes.

7 Troubleshooting

This section describes some troubleshooting techniques and the recommended method for stopping and starting nodes to prevent inadvertent data loss.

7.1 MariaDB Galera Cluster Status Variables

When MariaDB Galera Cluster is installed on a node, several special Galera-related status variables are set and updated. If you log on to a node with the mysql client, you can execute `SHOW STATUS` to see those variables. They all begin with `wsrep`, so you use the following command to show just those status variables:

```
SHOW STATUS LIKE 'wsrep%';

+----------------------------+--------------------------------------+
| Variable_name              | Value                                |
+----------------------------+--------------------------------------+
| wsrep_local_state_uuid     | bfd619ba-34cd-11e3-8744-22c71e172de0 |
| wsrep_protocol_version     | 4                                    |
| wsrep_last_committed       | 3                                    |
| wsrep_replicated           | 0                                    |
| wsrep_replicated_bytes     | 0                                    |
| wsrep_received             | 4                                    |
| wsrep_received_bytes       | 326                                  |
| wsrep_local_commits        | 0                                    |
| wsrep_local_cert_failures  | 0                                    |
| wsrep_local_bf_aborts      | 0                                    |
| wsrep_local_replays        | 0                                    |
| wsrep_local_send_queue     | 0                                    |
| wsrep_local_send_queue_avg | 0.000000                             |
| wsrep_local_recv_queue     | 0                                    |
| wsrep_local_recv_queue_avg | 0.000000                             |
| wsrep_flow_control_paused  | 0.000000                             |
| wsrep_flow_control_sent    | 0                                    |
| wsrep_flow_control_recv    | 0                                    |
| wsrep_cert_deps_distance   | 0.000000                             |
| wsrep_apply_oooe           | 0.000000                             |
| wsrep_apply_oool           | 0.000000                             |
| wsrep_apply_window         | 0.000000                             |
| wsrep_commit_oooe          | 0.000000                             |
| wsrep_commit_oool          | 0.000000                             |
| wsrep_commit_window        | 0.000000                             |
| wsrep_local_state          | 4                                    |
| wsrep_local_state_comment  | Synced                               |
| wsrep_cert_index_size      | 0                                    |
| wsrep_causal_reads         | 0                                    |
| wsrep_incoming_addresses   | 10.154.141.7:3306                    |
| wsrep_cluster_conf_id      | 3                                    |
| wsrep_cluster_size         | 1                                    |
| wsrep_cluster_state_uuid   | bfd619ba-34cd-11e3-8744-22c71e172de0 |
| wsrep_cluster_status       | Primary                              |
| wsrep_connected            | ON                                   |
| wsrep_local_index          | 0                                    |
| wsrep_provider_name        | Galera                               |
| wsrep_provider_vendor      | Codership Oy <info@codership.com>    |
| wsrep_provider_version     | 23.2.7(r157)                         |
| wsrep_ready                | ON                                   |
+----------------------------+--------------------------------------+
```

Notice the variable `wsrep_incoming_addresses`. It shows the IP addresses and ports of every active replication node in the system cluster. In the example result above there is only one address which means only one node is running or that that node only knows about itself and is not connected to any other nodes. On a system with several nodes running, they will all be shown in this variable in a comma-separated list.

This is the easiest way to quickly diagnose replication problems within a cluster — examine this variable to see the connection topology of a cluster and understand where expected connections are missing or incorrectly configured.

7.2 Time Synchronization

MariaDB Manager requires that all nodes have synchronized date and time to the management node. The security mechanism that prevents replay of messages relies upon this synchronization. All of the nodes in a system under the control of MariaDB Manager must be set to the same date and time and the same time zone. This common time setting also allows the cluster software to accurately determine which replica has the most up-to-date data, allowing the synchronous replication to operate correctly in maintaining a highly-available database cluster.

7.3 Starting Cluster Nodes in the Right Order

Galera Cluster synchronizes the nodes in a system to always have the latest data. If you create a new system with a single empty database node, that empty node, because it was started first, is deemed as having the latest data for the system. All other nodes that are subsequently added to the system synchronize their data with the current data of the system, without regard to any data they may already have. As a result, it is possible to inadvertently destroy the data in your database cluster. Because the first node added to a system determines the starting data, be sure to add nodes back into a system in last-stopped, first started order.

If you need to shut down all the nodes in a system, make note of which one you shut down last. In an active application environment, data will likely be added or changed in the time it takes to shut down all of the servers. In such a situation, the last server to shut down will have changes to the data that the other servers didn't receive through replication. Therefore, when starting the nodes up again, the last server to be shut down should be the first to be started. By following this procedure, all servers receive the changes they may have missed while they were shut down.

In the event that you wish to revert a system to a known earlier state, for example, in restoring a backup, you can take advantage of Galera's synchronization mechanism to replicate the earlier state over all the nodes in the system. First, isolate a node and restore the back-up to this node, then add that restored node to a new system as the first node. When you then rejoin the other nodes to this new system, Galera will consider the restored backup to be the most recent replica and overwrite the other nodes' data with the restored backup automatically.

7.4 Diagnosing and fixing issues with Installation and Provisioning

The following two subsections list installation and provisioning issues that have been encountered on the manager node and data nodes, and how to resolve them.

7.4.1 Potential manager node issues

Installation problems:

- *CentOS repositories are not available: not possible to install dependencies.* — To avoid this, make sure the management node, especially when MariaDB Enterprise Cluster is being installed, can access the Internet to be able to talk with both the MariaDB Enterprise Cluster and CentOS repositories.
- *3rd party repositories are configured: incompatible newer version of php, tomcat, etc are available (for example, newer php packages have different names).* — Before installing MariaDB Enterprise Cluster, check the repositories configured under `/etc/yum.repos.d/` and comment out or remove all third-party repositories other than the official CentOS repositories and the MariaDB Enterprise Cluster repositories. If incompatible versions of packages like php, tomcat, or java are installed, they may need to be uninstalled prior to installing MariaDB Enterprise Cluster.

Tomcat7 installation problems:

- *Tomcat is already installed to different place: possible conflict with tomcat start (8080 port is not available).* — A custom Tomcat installation may need to be removed before installing MariaDB Enterprise Cluster. If port 8080 is configured and being used by an alternative tomcat or other piece of software it will need to be stopped and possibly uninstalled.
- *Tomcat is installed from different RPM (currently the "MariaDB-Manager" repository contains tomcat7.x86_64 0:7.0.39-1 and MariaDB-Manager-API has strong dependency on 7.0.39-1): possible 'file conflict' or 'can't downgrade to 7.0.39-7), yum refuses to install tomcat.* — Tomcat packages from another repository other than the MariaDB Enterprise Cluster repository will need to be uninstalled before installing MariaDB Enterprise Cluster.
- *Tomcat is configured to use different directory (not `/usr/local/tomcat7/webapps/`) to store applications files.* — Existing Tomcat config files will either need to be removed prior to installing MariaDB Enterprise Cluster, or modified to use the `/usr/local/tomcat7/webapps/` directory.

Apache problems:

- *Apache is already installed and configured in incompatible way (for example, webroot directory is different from `/var/www/html/`).* — Existing Apache configuration files will need to be modified so that the webroot directory is `/var/www/html/`.

PHP problems:

- *PHP is already installed and the `php.ini` file is changed in a way which makes the command: `sed -i "s|;date.timezone =|date.timezone = $timezone|" /etc/php.ini` fail.* — PHP should be uninstalled and the customized `php.ini` file removed.

Java problems:

- *Currently, the MariaDB-Manager-WebUI package depends on `java-1.7.0-openjdk`: possible problem in case if some other java is installed.* — Incorrect versions of Java should be uninstalled prior to installing MariaDB Enterprise Cluster.

Security problems:

- *Wrong repository credentials in `/etc/yum.repos.d/MariaDB-Manager.repo`.* — Check the welcome letter from SkySQL for the correct credentials, and if they still don't work contact SkySQL.
- *Apparmor, SMACK, Tomoyo, or etc… is installed: php can't access administration DB, blocked ports, etc (MariaDB-Manager is able to disable only SELinux).* — These packages should be uninstalled prior to installing MariaDB Enterprise Cluster.

7.4.2 Potential data node issues

Installation problems:

- *3rd party repositories are configured: incompatible newer version of php, tomcat, etc are available (for example, newer php packages have different names).* — Before installing MariaDB Enterprise Cluster, check the repositories configured under /etc/yum.repos.d/ and comment out or remove all third-party repositories other than the official CentOS repositories and the MariaDB Enterprise Cluster repositories. If incompatible versions of packages like php, tomcat, or java are installed, they may need to be uninstalled prior to installing MariaDB Enterprise Cluster.
- If the openssh-clients and openssh-server packages are not installed on all nodes, the management node will be unable to provision the nodes. — See the System Requirements section for information on installing these packages and starting the OpenSSH server.

Security problems:

- *Access control system blocks root access: it blocks provisioning.* — Disable the access control system on the data node before provisioning the node.

7.5 Logs

MariaDB Manager uses `syslog` logging extensively. The logs are often the first place you should look if an error occurs as whatever went wrong will likely be recorded in one of the log files.

Logs can be searched for specific messages from each component, just as with other system services. For example, the most recent 1000 MariaDB Manager log messages on the management node can be viewed with the following command:

```
grep MariaDB-Manager-Task /var/log/messages | tail -n1000
```

This command, if the `messages` file on the management node is large, can be slow, so a way to retrieve a good portion of the most recent MariaDB Manager logs is to flip the command around like so:

```
tail -n1000 /var/log/messages | grep MariaDB-Manager-Task
```

This command grabs the most recent 1000 log messages and then searches those for messages from MariaDB Manager. Adjust `-n1000` according to your needs.

Other components can be searched in the same way. In addition to `MariaDB-Manager-Task`, the following words can be searched for:

- `MariaDB-Manager-API` — for API log messages
- `MariaDB-Manager-Monitor` — for Monitor log messages
- `MariaDB-Manager-Remote` — for Manager log messages on individual database nodes

The WebUI uses Apache Tomcat. Its logs can be found on the management node at:

```
/usr/local/tomcat7/logs
```

8 Appendix A: A Glimpse of the API

The MariaDB Enterprise Cluster Application Programming Interface (API) provides access to the managed systems through a range of REST-style requests. Adopting REST principles means that the API leverages the standards of the HTTP protocol. It also means that requests are made to URLs that refer to resources, with the action indicated by the HTTP request type. This differs from interfaces that follow remote procedure call (RPC) style, where URLs typically indicate actions.

The API can either be used by a front end such as a graphical user interface, or by other IT automation technologies in order to make such tools simpler and more robust. The MariaDB Manager WebUI, which is the default user interface for the product, is entirely implemented through the API.

The resources managed by the API are *Systems*, *Nodes*, *Monitors*, *Tasks*, *Schedules*, and *Backups*. Each is described briefly below.

A ***Node*** is a computer in a rather general sense. Often it may be a Virtual Private Server (VPS). Or it could be a blade in a blade server system. Or it could be a single machine in any configuration. Whatever its exact nature, it will be connected to a network so that it can be either a management node or a database node in a System.

A ***System*** is a collection of *nodes* that have some common purpose. Typically, a system is a group of database servers. Collectively, the group provides better availability or scalability than could be achieved by a single system. Multiple systems can be managed and provisioned from a single MariaDB Manager management node.

A ***Monitor*** defines something that can be observed, such as the volume of traffic reaching a Node. Successive observations are stored so as to create a record of the Node's behavior. Separate from the API, there is a constantly running program that carries out monitoring, but the observations to be captured are

defined by the Monitors known to the API, and the resulting data are stored by the API. The data can also be retrieved from the API.

A ***Task*** is the record of running a *command* on a Node. It includes details of what Command and what Node were involved, when the Command was instigated and when it completed (if it has). The step being run is recorded, along with the overall status and any error message.

A ***Schedule*** is information about a *command* that is to be run on a Node at some other time, possibly according to a recurring timetable. When a Schedule is run, it creates a Task resource so as to maintain a record of what has happened, and in the Task will contain a reference to the Schedule that triggered it. The Schedule can use an iCalendar Recurrence Rule (RRULE) to specify a repeating pattern of running a Command.

One *command* causes an extra resource to be created – a ***Backup***. Because information such as the location of the backup needs to be stored, the backup command causes the creation of this resource.

Because the API follows REST principles, operations are requested using HTTP requests. So, to obtain information about a particular system, the request would be an HTTP *GET*. It might be directed to a URL such as `http://example.com/path/to/api/system/3`. The response is a JSON object with a property of *system* that has a value of a system object.

To update the information about a particular system, the same URL would be used, but the request would be an `HTTP PUT`, while to delete the system the request would be `HTTP DELETE` and again the same URL. Because system ID numbers are allocated by the API, creating a system would be a request to `http://example.com/path/to/api/system` and the request would be `HTTP POST`.

Further information is conveyed from the requester to the API and in the API response by the use of HTTP headers. In particular, the success or otherwise of any request will be returned as the HTTP response code. For example, success is usually indicated by code `200`, whereas if a non-existent resource is requested, the code would be `404`. These codes may be familiar to you from using a browser.

The API is implemented in PHP and is run within a web server. The recommended web server is Apache.

More detail about the API is contained in the MariaDB Manager API Reference Manual at: https://mariadb.com/kb/en/mariadb-manager-api/

9 Appendix B: Example Galera Configuration

Following is an example Galera configuration file which may be used and modified if you are manually provisioning a database node. When using automatic provisioning (the default) creating your own configuration file is not necessary.

```
[mysqld]

# Row binary log format is required by Galera
binlog_format=ROW
log-bin

# InnoDB is currently the only storage engine supported by Galera
default-storage-engine=innodb
innodb_file_per_table

# To avoid issues with 'bulk mode inserts' using autoincrement fields
innodb_autoinc_lock_mode=2

# Required to prevent deadlocks on parallel transaction execution
innodb_locks_unsafe_for_binlog=1

# Query Cache is not supported by Galera wsrep replication
query_cache_size=0
query_cache_type=0

# INITIAL SETUP
# In some systems bind-address defaults to 127.0.0.1, and with
# mysqldump SST it will have (most likely) disastrous consequences
# on donor node
bind-address=10.199.33.198

##
## WSREP options
##

# INITIAL SETUP
# For the initial setup, wsrep should be disabled
wsrep_provider=none
# After initial setup, parameter should have full path to wsrep
# provider library
wsrep_provider=/usr/lib64/galera/libgalera_smm.so

# Provider specific configuration options
wsrep_provider_options = "evs.keepalive_period = PT3S;
evs.inactive_check_period = PT10S; evs.suspect_timeout = PT30S;
evs.inactive_timeout = PT1M; evs.install_timeout = PT1M"

# Logical cluster name. Should be the same for all nodes in the same
cluster.
wsrep_cluster_name=skycluster

# INITIAL SETUP
# Group communication system handle: for the first node to be launched,
# the value should be "gcomm://", indicating creation of a new cluster;
# for the other nodes joining the cluster, the value should be
```

```
# "gcomm://xxx.xxx.xxx.xxx:4567", where xxx.xxx.xxx.xxx should be the
# ip of a node already on the cluster (usually the first one)
wsrep_cluster_address=gcomm://

# Human-readable node name (non-unique). Hostname by default.
wsrep_node_name=Node 2

# INITIAL SETUP
# Base replication <address|hostname>[:port] of the node.
# The values supplied will be used as defaults for state transfer
# receiving, listening ports and so on. Default: address of the
# first network interface.
wsrep_node_address=10.199.33.198

# INITIAL SETUP
# Address for incoming client connections. Autodetect by default.
wsrep_node_incoming_address=10.199.33.198

# Number of threads that will process writesets from other nodes
wsrep_slave_threads=1

# Generate fake primary keys for non-PK tables (required for multi-master
# and parallel applying operation)
wsrep_certify_nonPK=1

# Maximum number of rows in write set
wsrep_max_ws_rows=131072

# Maximum size of write set
wsrep_max_ws_size=1073741824

# Debug level logging (1 = enabled)
wsrep_debug=1

# Convert locking sessions into transactions
wsrep_convert_LOCK_to_trx=0

# Number of retries for deadlocked autocommits
wsrep_retry_autocommit=1

# Change auto_increment_increment and auto_increment_offset automatically
wsrep_auto_increment_control=1

# Retry autoinc insert, when the insert failed for "duplicate key error"
wsrep_drupal_282555_workaround=0

# Enable "strictly synchronous" semantics for read operations
wsrep_causal_reads=0

# Command to call when node status or cluster membership changes.
```

```
# Will be passed all or some of the following options:
# --status  - new status of this node
# --uuid- UUID of the cluster
# --primary - whether the component is primary or not ("yes"/"no")
# --members - comma-separated list of members
# --index   - index of this node in the list
wsrep_notify_cmd=

##
## WSREP State Transfer options
##

# State Snapshot Transfer method
#wsrep_sst_method=mysqldump
wsrep_sst_method=xtrabackup

# INITIAL SETUP
# Address which donor should send State Snapshot to.
# Should be the address of the CURRENT node.
# DON'T SET IT TO DONOR ADDRESS!!!
# (SST method dependent. Defaults to the first IP of the first interface)
wsrep_sst_receive_address=10.199.33.198

# INITIAL SETUP
# SST authentication string. This will be used to send SST to joining
nodes.
# Depends on SST method. For mysqldump method it is root:<root password>
wsrep_sst_auth=repluser:repluser1

# Desired SST donor name.
#wsrep_sst_donor=

# Reject client queries when donating SST (false)
#wsrep_sst_donor_rejects_queries=0

# Protocol version to use
# wsrep_protocol_version=
```