w: www.mariadb.com



e: info@mariadb.com

MariaDB Enterprise – Getting Started Guide (Beta)

Table of Contents

What is MariaDB Enterprise?	2
Introducing MariaDB Manager	3
Installing MariaDB Enterprise	3
System Requirements	3
Installing MariaDB Manager	4
Configuring Systems	5
The Initial Start	5
Adding a System	5
Adding and Configuring Nodes	7
Adding a Node	7
Control and Commands	8
Initializing a Node	9
Starting a Node	10
Node Management	12
MariaDB Galera Cluster Variables	12
Isolating, Backing Up, and Restoring a Node	13
Last Stopped, First Started	14

What is MariaDB Enterprise?

MariaDB Enterprise is a set of management tools and an API with which we can easily provision, monitor, and manage a highly available cluster of MariaDB server nodes configured to use the Galera library for multi-master, synchronous replication. Galera is a powerful technology that can eliminate single points of failure for our database infrastructure, but it is relatively new and can be a challenge to configure for administrators who aren't familiar with it.

MariaDB Enterprise 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 includes three main components:

- MariaDB Manager, a GUI interface to the monitoring and administration features of the product,
- The MariaDB Manager API, the central hub that provides the provisioning, monitoring, and management capabilities through a RESTful API,
- MariaDB Enterprise Monitor, a data collection engine that connects to the database nodes in a MariaDB Galera Cluster, stores and analyzes statistics on operation, and delivers data to the API engine.

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



Introducing MariaDB Manager

MariaDB Manager is a graphical web user interface (WUI) that utilizes the MariaDB Manager API to interact with servers running MariaDB Galera Cluster. Following is a screenshot of this web interface. With it, we can install multiple systems containing sets of replication servers or nodes running MariaDB Galera Cluster. It can also install MariaDB Galera Cluster on the nodes within systems — making installation and configuration much easier. It acts as a facilitator for MariaDB Galera Cluster servers, allowing us to view information on each system and each node, setup replication between servers, stop servers, isolate servers from others to make clean backups, and bring stopped servers back online.



Installing MariaDB Enterprise

Installation of MariaDB Enterprise can be performed manually by downloading the individual RPMs and installing them on our management server and on each replication server. However, the process can be greatly simplified using MariaDB Manager. We just need to install it on our management server, and it will handle the installation of MariaDB Galera Cluster and all other required software on all of our replication servers. To make it even simpler, we can install MariaDB Manager from the SkySQL yum repository. We just need to make sure the servers we are using meet the system requirements.

System Requirements

MariaDB Manager is the main program through which we will manage our MariaDB Galera Cluster servers. We need to install it on a server other than the servers which will host our databases. There are only a few prerequisites for installing MariaDB Manager on this server.

Our 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.

At this stage in the development process, MariaDB Enterprise should only be installed on a 64-bit server running CentOS 6.x. There is an installation for 32-bit systems, but there are problems with Tomcat related to it. This will be resolved soon. For now, use only 64-bit systems.

MariaDB Manager needs to have root access to the server on which it will operate. Once this main server is in place, we can proceed to installing our replication nodes.

Installing MariaDB Manager

To make installation easier and simpler, the MariaDB website has a repository for installing MariaDB Manager. To add this repository we create a file under /etc/yum.repos.d/ (i.e.,

/etc/yum.repos.d/MariaDB-Manager.repo). We can create manually this file with a simple text editor or by downloading a ready-made file. Following is an example of the file. It is recommended that we download the latest version directly from the website, as the contents may have changed.

[skysql] name=skysql baseurl=<u>http://user:pwd@code.mariadb.com/MariaDB-Manager/beta/repo/x86_64</u> enabled=1 gpgcheck=false [skysql-noarch] name=skysql-noarch baseurl=<u>http://user:pwd@code.mariadb.com/MariaDB-Manager/beta/repo/noarch/</u> enabled=1 gpgcheck=false

The first section of this configuration file is for the 64-bit-only packages and the second part is for packages that can be used on 64-bit and 32-bit systems. This in preparation for when 32-bit systems are supported. Note that required packages for 32-bit systems do not exist at this time. When they are, a 32-bit-only section will be created for those packages similar to the 64-bit-only section.

As mentioned previously, instead of creating this configuration file by hand, we should download the latest version directly from the MariaDB website. We 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 our management server (replacing user:pwd with our 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 \
<u>http://code.mariadb.com/MariaDB-Manager/beta/MariaDB-Manager-beta.repo</u> \
> /etc/yum.repos.d/MariaDB-Manager.repo

Once the repository configuration file is in place, we will need to edit it to add our username and password to the file. The file from the MariaDB website has placeholders for our username and password on the *two* baseurl lines. We just need to open the file in a text editor and replace the user:pwd text with our username and password, separated by a colon (:). After adding our username

and password, we can use yum to install MariaDB Manager and related software. To do so, run the following command as the root user:

yum install MariaDB-Manager

Our system will be checked for any dependencies and present us with a list of packages to install. Unless the system asks us to uninstall something that we need for other services we have running on the server, or it wants to upgrade software that might cause problems for us, it is generally safe to accept the list of recommendations. Once yum has finished installing everything, we can proceed to using MariaDB Manager to configure our setup, and to install and connect to our replication servers.

Configuring Systems

Configuring a system with MariaDB Manager is fairly straightforward. We login from a web interface, add systems, and then add nodes within each system. Installation of software on the nodes and configuration of that software are handled automatically. Without MariaDB Manager, this would be a time consuming and possibly frustrating process. MariaDB Manager makes the process quick and easy.

The Initial Start

We will typically install MariaDB Manager on a server which will not have MariaDB running on it. The databases will be located on other servers which we will setup

and provision using MariaDB Manager.

After installing MariaDB Manager on a server, the web-based user interface will be running and we can access it using a web browser (Firefox is recommended). For the address, enter the 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.xxx.xxx.8080/MariaDBManager

If MariaDB Manager was installed successfully, the first time we start the web user interface, we will be asked to provide a username and password for it to use to create the primary

administrator account. Do this immediately after installation as until this user is setup our installation is vulnerable.

Adding a System

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



Add a System	
Name	Sys1
Туре	galera ‡
Database Username*	db_user
Database Password*	•••••
Confirm Password*	•••••
Replication Username*	replication_user
Replication Password*	•••••
Confirm Password*	•••••
Replication Username* Replication Password* Confirm Password*	replication_user

In the box labeled, *Name*, we enter the name we want to use to refer to our system. Any name is fine, but it should be something logical (and preferably short) that we will recognize and remember in relation to the set of servers we will be adding to the system.

The next field is *Type*. At this time, the only choice available is *galera*. So select it.

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

When we've finished filling in the information about the system, click the *Add System* button. We should see quickly afterwards an icon similar to the one on the right appear in the MariaDB Manager interface.



Remember: a system in this sense is not a server, but it's rather a collection of nodes. Click the button, *Done* to complete this part of the process.

Our next step is to add nodes. A system with no nodes is not very useful. The next section describes how to add nodes and to install and setup MariaDB Galera Cluster on those nodes.

Adding and Configuring Nodes

Now that we've added the first system, we're ready to install nodes that will be part of that system. Nodes are servers on which MariaDB and Galera will be installed and will include eventually data. We can add existing servers which have already the needed software, or add new servers on which the MariaDB Manager will install all of the needed software. The process is fairly simple. After we've added a node, we will be able to start the node from MariaDB Manager, monitor its activities, and run various administrative commands.

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. Once we've done that, if we haven't yet added any other nodes to the system, there will be a message on the screen that says, *No Components Available*. Click on the button, *Add Node* to add a node to the system. When we do, we'll see a dialog box like the one below:

Add a Node to the S	System: Sys1	
Name	Node1	
Hostname		
Instance ID		
Public IP		
Private IP*	10.10.0.2	
i Connection o	options	
Node is a	vailable now, cor	nect automatically.
Poot Password	bi avaitable, user	
ROOL Password		
Confirm Password	•••••	

There are several values that we may enter in this box, but there are only a couple that are required. First we need to give the node a name we will recognize. This is for all users who may access the system, not for the server to use, so it does not have to match the server name or DNS entry. It should, however, be something logical. The other required field is the *Private IP address*. This is the internal IP address of the server within the same private network as the server hosting MariaDB Manager. The *Public IP Address* is the address by which the public will access the data server, but not the one MariaDB Manager will use. In future releases of MariaDB Manager, we won't be required to enter a private IP address and may use instead only a public IP address. For now, the *Public IP Address* field can be left blank. The *Hostname* and *Instance ID* fields may also be left blank at this point. If the server is up and running, select the *Node is available now, connect automatically* radio-button. If not, select the *Node is not available, user will run connect later* radio button and once the node is up and running we will be able to manually start the initial connection and setup process.

Last we enter the root password for the node. Because MariaDB Manager will be handling the installation of MariaDB Galera Cluster and related packages, we don't need to log into ourselves. MariaDB Manager will use the default values we set when creating the system. However we need to enter the same root password or SSH key that we gave when creating the system. In a future release of MariaDB Manager, when selecting *galera* as the type of system, these fields will be filled in with the default information and grayed to indicate they cannot be changed at this level.

There are plans to allow a different user account instead of root to be given and its password in the future.

When we're finished entering information about the new node, click the *Add Node* button. Then click the button, *Done*. We should then see an icon similar to the one here on the right. The name of the node in this icon is Node 1, since that was how it was named when created. Our screen will show whatever name we gave for the node. If we click on that icon, the main panel will provide information on node. However, since the node hasn't been installed yet, there's nothing much to see. To finish creating the node and to get it started, click on the *Control tab*.



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, we click on it and then the large green triangle-shaped *run* button. To stop a command that has started, there is a large red square *stop* button. When not available, the colors of these buttons are lighter (in the following screenshot the *run* button is not available, and the *stop* button is. When a command is running, to the right of these buttons will be a graphical line showing the progress of the steps involved. Some actions involve 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 before a process has started; yellow for in progress; red for failed; and green for completed successfully. To the right of all of this is a general comment on the progress of the command. Refreshing the browser will update the comment and not affect the processes that are running or those that have completed.



Initializing a Node

After we've created a new node, click on the *Control* tab for the new node and we should see something similar to the previous screenshot. The only command available will be *connect*. It starts automatically when a node is added, so we don't have to click on that command to start it. To the right of the command and start button we will see steps related to starting the node for the first time.



During this initial connect phase, MariaDB Manager will check to see if the IP address and other information provided are correct, register information about the new node, and install a small script and create users for itself. All of these tasks enable it to interact with the data server when running all subsequent tasks. Incidentally, the icon for the node will change a bit while this step is running; it will have the word, *Connect* written across it, like the icon on the right.

If the MariaDB Manager cannot connect to the node, the icon for the node will have a red triangle shaped warning sign. If we hold our mouse pointer over the image, we will see a warning message similar to the one below. The message will explain the problem with details on what went wrong so we can fix it.



After 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*. Following is a screenshot showing how that will look in the *Control* tab. Click on *probe* and then click the run button to start the command. The MariaDB Manager will then attempt to probe the node to ensure that the data server can support MariaDB and other related software. It will look to see if MariaDB and other software are already installed and running. This way it won't overwrite existing databases.

Info	Control		
Comma probe	ands	probe probe	Has not run yet

If MariaDB or MySQL is installed and running on the new node, MariaDB Manager will ask us if we want to make it part of the system. Be careful at this point if we already have data on the node. If this is not our first node in the system the data on it might be overwritten by the data on already existing nodes.

If MariaDB Manager determines that the new node does not have MariaDB or related software installed on it, it will install the needed software on the node. There is no need to download or install manually the software. This process is known as *provisioning*.

When MariaDB Manager is installed, it uses the repositories for which it was configured to obtain in advance the software needed to install on the nodes. With each node installation, it uses the same RPM packages so that all of the nodes are using the same versions of the software and are therefore compatible with each other. This can also potentially make for faster provisioning of new nodes.

Starting a Node

Once we've 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.

Info Control								
Comprainds provident	Install the packager	provision packages configure	Running g the new node					
Previously run Commands								
Previously run Commands Started	Completed	Command	Parameters	Steps	PID	Private IP	User	State
Previously run Commands Started Mon, 14 Oct 2013 13:23:53 ·	Completed -0000 Man, 14 Oct 2013 13:23:53	Command -0000 provision	Parameters	Steps Install-packages,configure	PID 4218	Private IP 10.208.175.5	User Russell Dyer	State
Previously run Commands Started Mon, 14 Oct 2013 13:23:53 Mon, 14 Oct 2013 13:23:03	Completed +0000 Man, 14 Oct 2013 13:23:53	Command +0000 provision +0000 probe	Parameters	Steps Install-packages, configure probe	PID 4218 4156	Private IP 10.208.175.5 10.208.175.5	User Russell Dyer Russell Dyer	State Running Done - normal ci

If for some reason the node fails to start, the icon for the node will change to a red version like the one on the right. Additionally, we will see a message on the screen which says, *Failed to Launch*, along with other information to indicate what was accomplished and what wasn't.





There are many reasons that could cause a node to fail to start. Try clicking 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 work, we can check the system logs on the node's server for clues as to the

problem. If we think we may have entered something incorrectly when creating the node, we can delete it from MariaDB Manager and then recreate it.

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 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 we choose to delete the node, a dialog box will appear to ask us if we're sure that we want to delete the node. If we delete the node, it will be removed from the list and will no longer receive updates. After deletion, the underlying server will still be there along with any data it had prior to being deleted.



Note: If we entered the IP address of a node incorrectly when we created it and we want to delete and then recreate the node, we'll need to edit the /etc/my.cnf.d/skysql-galera.cnf file on the node we are re-provisioning to fix the IP address.

If we don't have problems starting a node, then the icon will look a bit green like the icon on the right in the following screenshot. In the box containing the list of commands, we will have several commands available now. These include *stop*, *restart*, *isolate*, *backup*, and *restore*. The *stop* and *restart* commands are fairly self explanatory. We'll go into the *isolate*, *backup*, and *restore* commands in the next section. All of these commands require the node to be running so if we *stop* the node, they will all be removed and we'll have only *start* available again. For some systems, it's better to *stop* the node and then *start* it again instead of using the *restart* command.

Systems		Components
0.0		
Sys1		Node1
Info Control		
Commands		
stop restart isolate backup restore	No Command is currently running on this node	

Node Management

There are a few things we need to be aware of when managing systems, and nodes within those systems, in MariaDB Manager. First, for resolving problems and understanding the system better, we should be aware of the special MariaDB Galera Cluster variables. To protect data, we should also know the process by which we make backups of nodes. Lastly, to prevent loss of data during general node maintenance, we need to familiarize ourselves with the proper method for stopping and starting nodes.

MariaDB Galera Cluster Variables

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

SHOW STATUS LIKE 'wsrep%';
++
Variable_name Value
++
wsrep_rotocol_version 4
wsrep_last_committed 3
wsrep_replicated 0
wsrep_replicated_bytes 0
wsrep_received 4
wsrep_local_commits 0
wsrep_local_cert_failures 0
wsrep_local_bf_aborts 0
wsrep_local_replays U
wsrep_local_send_queue_avg 0.000000



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.

Isolating, Backing Up, and Restoring a Node

Once we have several nodes installed and running with data on them, if we want to make a backup of a node, it's best first to *isolate* the node from the other nodes. This way it is unaffected by data changes from the others, and its activities won't affect the other nodes. Plus, users won't interact with the node while we're making the backup. Once a node has been isolated, then we can click on *backup* to make a backup of the databases on the node. If we run the *backup* command without running the *isolate* command first, MariaDB Manager will automatically run the *isolate* command for us before running the *backup* command.

Syste	ems		Components
0.0			
Sys1			Node1
(Cantral		
Info	Control		
Comma	inds		
stop			
isolate			
backup		No Command is currently running on this node	

When the backup is finished, we can then click on rejoin to have the isolated server *rejoin* the mix of nodes for the system. The other nodes that have been active will then provide snapshots to the node that is now rejoining the system to get it in synch and up-to-date with the others — not SQL statements like normal MySQL replication would provide.

For when we want to *restore* a backup, we would first *isolate* the node on which we want to restore data. Then execute the *restore* command to restore the data. If we have multiple nodes, though, we may want to *isolate* the other nodes before we *rejoin* the restored node. Otherwise, the other nodes will assume that the joining node has old data and provide it with snapshots which will overwrite the restored data. Which node is assumed to be the current one is based on time settings. It is a requirement of the MariaDB Manager API security that all servers have synchronized date and time. The security mechanism that prevents replay of messages relies upon this. Therefore, it's best that all of the nodes are set to the same date and time and are set to the same time zone. They can be located anywhere, but they should all be set to the same time. In a future release of MariaDB Manager, times will be adjusted and compared based on UTC and time zones. Then we won't have to set all of the nodes to the same time.

Last Stopped, First Started

Once we have several nodes installed and running with data on them, we may have a need to shut down all the nodes in a system. If we shut down all of the nodes, we should make note of which one we shut down last. If the nodes are very active, 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. Therefore, when starting the nodes up again, the last server to be shut down should be the first to be started. That way the other servers get the changes they missed while they were shut down.