MySQL InnoDB Cluster – Extended

Last Updated: March 2024

Contents

Overview

Purpose

The purpose of this document is to provide information on supporting MySQL InnoDB Cluster and its components. This document will cover the components that make up the solution, installation of the components and commands to setup, support and troubleshoot the environment.

General

InnoDB Cluster features one primary read-write node alongside several secondary nodes designated for read-only operations. While it's technically feasible to set up a MySQL Cluster where all nodes are configured for read-write operations, both MySQL and the Morpheus PS team advise against it.

InnoDB Cluster maintains synchronous replication among its members within the same site and asynchronous replication between sites. In the event of the primary node failure, InnoDB automatically promotes another node within the same site to assume the primary role.

In the event of a site outage within a multi-site environment, you would initiate an emergency failover process from MySQL Shell.

Components

MySQL

The MySQL software delivers a very fast, multithreaded, multi-user, and robust SQL (Structured Query Language) database server. MySQL Server is intended for mission-critical, heavy-load production systems as well as for embedding into mass-deployed software. MySQL will be the foundational relational database used by Morpheus.

For additional information on MySQL 8:
https://dev.mysql.com/doc/refman/8.0/en/introduction.html

MySQL Shell

MySQl Shell is an advanced client and code editor for MySQL. In addition to the provided SQL functionality, MySQL Shell provides scripting capabilities for JavaScript and Python and includes APIs for working with MySQL. X DevAPI enables you to work with both relational and document data. AdminAPI enables you to work with InnoDB Cluster, InnoDB ClusterSet, and InnoDB ReplicaSet.

MySQL Shell is what is used to maintain and configure the InnoDB cluster. You will find common commands that are necessary to manage InnoDB cluster further in this guide.

For additional information on MySQL Shell:
https://dev.mysql.com/doc/mysql-shell/8.0/en/

MySQL Router

MySQL Router is lightweight middleware that provides transparent routing between your application and any backend MySQL Servers. This router can run directly on the Morpheus application nodes and be bootstrapped from the MySQL cluster.

MySQL Router is responsible for routing Morpheus application node traffic to the primary database read-write node.

For additional information on MySQL Router:
https://dev.mysql.com/doc/mysql-router/8.0/en/

Installation

MySQL

You can install MySQL from the OS repositories or add the MySQL community repository. There are no additional 3rd party installations required for an InnoDB Cluster.

Using OS repositories:
Check for available versions from the repo.

apt - apt-cache show mysql-server
yum - yum list mysql-server

If the version is 8.0.32 or greater then install.

apt - apt-get update
- apt-get -y install mysql-server

yum - yum update
- yum -y install mysql-server

Using the MySQL Repos.

- yum - https://dev.mysql.com/doc/refman/8.0/en/linux-installation-yum-repo.html
- apt - https://dev.mysql.com/doc/refman/8.0/en/linux-installation-apt-repo.html

Configuration – After installation you will need to configure a few settings for use with the cluster.

Set the root password and create an admin account you will use as your Cluster Admin.

- ALTER USER 'root'@'localhost' IDENTIFIED WITH caching_sha2_password BY 'password';
- CREATE USER 'clusterAdmin'@'%' IDENTIFIED BY 'password';
- GRANT ALL PRIVILEGES ON *.* TO 'clusterAdmin'@'%' with grant option;

Set to generate invisible primary keys (InnoDB Cluster requires all tables to have primary keys. This is required to allow Morpheus to run on an InnoDB Cluster.

- set persist sql_generate_invisible_primary_key=1;

Add the following to the my.cnf

- innodb_buffer_pool_size=6G #This should be set to a size of 70 – 80% of the installed RAM
- innodb_buffer_pool_instances=6 #Ths should be set to one instance per gig of the pool size
- innodb_use_fdatasync=ON # This changes the sync method for better performance.
- bind-address=0.0.0.0 # This is binding MySQL to all interfaces.

Restart the MySQL service.

- systemctl restart mysql or systemctl restart mysqld

MySQL Shell

MySQL Shell can be installed on any machine you want to manage InnoDB cluster from. We will typically install this on each Morpheus application node to have it available for setup and management during the install and post deployment.

MySQL Shell will be used for but not limited to checking cluster health, removing/adding site members, manual failover between site members, failover between sites.

Manual Install

- go to https://dev.mysql.com/downloads/shell/ and download the correct installer for your Linux distro.
- Install the package.

From repo

- Follow these steps to add the repo. https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-shell-install-linux-quick.html
- Then use apt-get install mysql-shell or yum install mysql-shell

MySQL Router

MySQL Router will be installed on each Morpheus Application node. This will be the middleware that monitors the database cluster and points the Morpheus application node to the correct Primary Database node. Morpheus will connect to it on 127.0.0.1 and the R/W default port of 6446.

Manual Install

- go to https://dev.mysql.com/downloads/router/ and download the correct installer for your Linux distro.
- Install the package.

From Repo

- Follow these steps to add the repo. https://dev.mysql.com/doc/mysql-router/8.0/en/mysql-router-installation-linux.html
- Then use apt-get install mysql-router or yum install mysql-router

Configuration

- Make sure to create the router user from MySQL Shell before setting up MySQL Router. Otherwise, the user will be created with the legacy native plugin.
  - \c clusterAdmin@dba-1:3306
  - dba.getCluster().setupRouterAccount('routeruser')
- Run the following to bootstrap the InnoDBCluster. With this config MySQL Router will get its config from the cluster and update based on the primary node that is active across any site in the cluster set.
  - mysqlrouter --bootstrap clusterAdmin@adb-5:3306 --account routeruser --user=mysqlrouter --disable-rest
- Restart the router for changes to take effect.
  - systemctl restart mysqlrouter
- Confirm the router is listening on the ports.
  - sudo lsof -i -P -n | grep LISTEN | grep mysqlrout

MySQL Shell Commands

Start MySQL Shell

```
mysqlsh
```

Exit MySQL Shell

```
\exit
```

Save History

```
\option --persist history.autoSave=1
```

Connect to a cluster node

```
\c clusterAdmin@dba-1:3306       #dba-1 is the db server name in this example
```

Get the Clusters Status

```
\c clusterAdmin@dba-1:3306      #dba-1 is the db server name in this example
cluster = dba.getCluster()
cluster.status()
```

Get the Clusters Extended Status

```
\c clusterAdmin@dba-1:3306      #dba-1 is the db server name in this example
cluster = dba.getCluster()
cluster.status({extended: 1})
```

Get the Cluster Set Status (For Multi Site)

```
\c clusterAdmin@dba-1:3306      #dba-1 is the db server name in this example
clusterset = dba.getClusterSet()
clusterset.status()
```

Get the Cluster Set Extended Status (For Multi Site)

```
\c clusterAdmin@dba-1:3306      #dba-1 is the db server name in this example
clusterset = dba.getClusterSet()
clusterset.status({extended: 1})
```

Check if DB Node is Ready for Configuration.

```
dba.checkInstanceConfiguration('clusterAdmin@dba-1:3306')
```

Configure DB Node to Meet Requirements for Clustering

```
dba.configureInstance('clusterAdmin@dba-1:3306')
```

Create a Single Site Cluster

Make sure you have run the “Configure DB Node to Meet Requirements” on each of the Cluster nodes first.

```
\c clusterAdmin@dba-1:3306        #Connect to one of the cluster nodes
cluster = dba.createCluster("A")     # Create the cluster. “A” is the cluster name in this example
cluster.addInstance("dba-2:3306")   #Add the second node to the cluster.
cluster.addInstance("dba-3:3306")   #Add the third node to the cluster.
cluster.status()                          #Get the cluster status to confirm functional
```

Create a Multi Site Cluster

Make sure you have run the “Configure DB Node to Meet Requirements” on each of the Cluster nodes first.

```
\c clusterAdmin@dba-1:3306          #Connect to one of the cluster nodes
cluster = dba.createCluster("A")    # Create the cluster. “A” is the cluster name in this example
cluster.addInstance("dba-2:3306")   #Add the second node to the cluster.
cluster.addInstance("dba-3:3306")   #Add the third node to the cluster.
clusterset = cluster.createClusterSet("ClusterSet")  #Create the clusterset “ClusterSet”
clusterb = clusterset.createReplicaCluster("dbb-1:3306", "B")  Create the replica cluster “B”
clusterb.addInstance("dbb-2:3306")   #Add the second node to Cluster “B”
clusterb.addInstance("dbb-3:3306")   #Add the third node to Cluster “B”
clusterset.status()                  #Get the cluster set status and confirm functionality
```

Remove a Node from the Cluster

```
\c clusterAdmin@dba-1:3306          #Connect to one of the cluster nodes
cluster = dba.getCluster()    # Get the cluster
cluster.removeInstance('clusterAdmin@dba-2:3306')   #We are removing the node dba-2
cluster.status()      # confirm status
```

Add a Node to the Cluster

```
\c clusterAdmin@dba-1:3306          #Connect to one of the cluster nodes
cluster = dba.getCluster()    # Get the cluster
cluster.addInstance('clusterAdmin@dba-2:3306')   #We are adding the node dba-2
cluster.status()      # confirm status
```

Run a Script

```
mysqlsh --file myscript.js
```

Planned Failover to Another Cluster Member at the Same Site

```
\c clusterAdmin@dba-1:3306          #Connect to one of the cluster nodes
cluster = dba.getCluster()    # Get the cluster.
cluster.setPrimaryInstance("dba-2:3306") #We are failing to a member that is not the primary
cluster.status()      # confirm status
```

Planned Failover to Another Site

```
\c clusterAdmin@dba-1:3306          #Connect to one of the cluster nodes
cs = dba.getClusterSet()    # Get the Cluster Set.
cs.setPrimaryCluster("B")   #B is our second site in this example
cs.status()      # confirm status
```

All Nodes Down Recovery

```
\c clusterAdmin@dba-1:3306          #Connect to one of the cluster nodes
dba.rebootClusterFromCompleteOutage()
```

Emergency Failover When a Site is Down

This should only be done if Primary site cannot be brought up first. Data loss could occur from Point in time of last replication.

```
\c clusterAdmin@dbb-1:3306          #Connect to one of the cluster nodes at the failover site.
clusterset = dba.getClusterSet()    #Get Cluster Set.
clusterset.status()                 #Check the Status. This will show you the site names.
clusterset.forcePrimaryCluster("B")   #This will force failover to site “B”.
```

Emergency Failover Recovery of Down site.

Once power is restored to Site A nodes. You can go through this repair process.

```
\c clusterAdmin@dba-1:3306            #Connect to one of the cluster nodes at the primary site.
dba.rebootClusterFromCompleteOutage()    #Recover from all the nodes being down.
clusterset = dba.getClusterSet()         #Get the cluster Set.
clusterset.rejoinCluster("A")          #Rejoin the Cluster “A”
```

Upgrades

https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-innodb-cluster-upgrade-rolling.html

Upgrade Order

MySQL Router
MySQL Shell
MySQL Server

Upgrade MySQL Router

MySQL Router should be installed on each of your Morpheus Application Nodes. You should upgrade one node at a time and confirm its up before moving to the next node in a rolling fashion.

- Download the package manually or use your added repo from the MySQL Router section above.
- systemctl stop mysqlrouter
- upgrade the package
- systemctl start mysqlrouter
- confirm MySQL Router is listening on 6446 and 6447

Upgrade MySQL Shell

It is best to always be running the latest version of MySQL Shell even if your version of MySQL is older. MySQL Shell should be installed on all Morpheus Application Nodes.

- Download the package manually or use your added repo from the MySQL Shell section above.
- upgrade the package
- confirm you can run MySQL Shell mysqlsh

Upgrade MySQL

MySQL Should be installed on at least 3 Database nodes external from the Morpheus Application Nodes. You should upgrade the secondary nodes in the cluster first and then the Primary node.

- Confirm which nodes are Secondary and Primary with mysqlsh
  - \c clusterAdmin@dbb-1:3306   #Connect to a node
  - cluster = dba.getCluster()
  - cluster.status()
- Download the package manually or use your added repo from the MySQL section above.
- Stop the MySQL Service
  - systemctl stop mysqld or systemctl stop mysql
- Upgrade MySQL
- Start the MySQL Service
  - systemctl start mysqld or systemctl start mysql

Post Upgrade Tasks

Check the cluster status after all upgrades
\c clusterAdmin@dbb-1:3306 #Connect to a node
cluster = dba.getCluster()
cluster.status({extended: true}) # get the extended status
Look to see if there are any errors listed for any of the nodes.
If any errors reported run the commands stated in the status to fix them.

Metadata Upgrade

After a MySQL upgrade you may need to upgrade the clusters metadata.

- \c clusterAdmin@dba-1:3306 #Connect to one of the cluster nodes.
- dba.upgradeMetadata() #This will upgrade the metadata.

Upgrade Morpheus in a Multi-Site Environment

This guide assumes you are following the general recommended architecture for a multi-site deployment.

Other than while upgrading Morpheus Application nodes, the morpheus-ui service should be down at the secondary site to make sure users are not accidentally using both sites simultaneously, as this will lead to issues.

To check if the morpheus-ui service is down on the application nodes you can run **morpheus-ctl status morpheus-ui** from the command line.

Environment

- Site A (Primary) and Site B (Secondary)
- 3 Application nodes and 3 Database nodes at *each* site
- morpheus-ui service is down at the secondary site.

Process

- Upgrade Site A following the standard HA environment upgrade process
- If required, conduct a failover of the database to Site B
- Upgrade Site B following the standard HA environment upgrade process
- Stop the morpheus-ui service on appnodes at Site B if it is running
    - morpheus-ctl stop morpheus-ui
- If required, failback the database to Site A

Performing a database failover to the secondary site may be necessary in the upgrade process under certain circumstances, such as when latency is too high between the two sites or when application nodes at secondary site lack access to the primary sites database nodes.

Morpheus Appliance Backup Settings

MySQL settings for MySQL Dump on Morpheus Application Nodes

Create the following file on the Morpheus Application nodes. This will set the proper setting for a MySQL dump db backup using InnoDB Cluster.

/etc/my.cnf

Add the following config to my.cnf

[mysqldump]
set-gtid-purged=OFF