MySQL InnoDB Cluster

New Feature in MySQL >= 5.7.17

Sergej Kurakin

Sergej Kurakin

Age: 36

Company: NFQ Technologies

Position: Software Engineer

Problem

What if your database server fails?

Reboot? Accidental shutdown?

Kernel panic? Misconfiguration?

Network failure? Hardware failure?

And this is the ONLY ONE database server that exists for your project.

Visitors, clients and business can’t wait until someone eventually gets to the Data Center and solves all issues.

Of course, there are solutions already created to automate failover and some of them are quite old and effective!

Master-Slave Replication with manual or automatic failover.

Linux High Availability Cluster with DRBD and MySQL.

MySQL NDB Cluster.

Oracle MySQL Cloud Service.

MySQL with Solaris Cluster.

Galera Cluster for MySQL.

Amazon Relational Database Service.

But we will talk about the new one!

MySQL InnoDB Cluster

MySQL InnoDB Cluster was introduced in version 5.7.17 on December 12, 2016.

Actually, MySQL Group Replication was introduced in 5.7.17 release and made MySQL InnoDB Cluster possible.

MySQL InnoDB Cluster is a collection of products that work together to provide a complete High Availability solution for MySQL.

Products used to build InnoDB Cluster

MySQL Shell >= 1.0.9

MySQL Router >= 2.1.3

MySQL Server >= 5.7.17

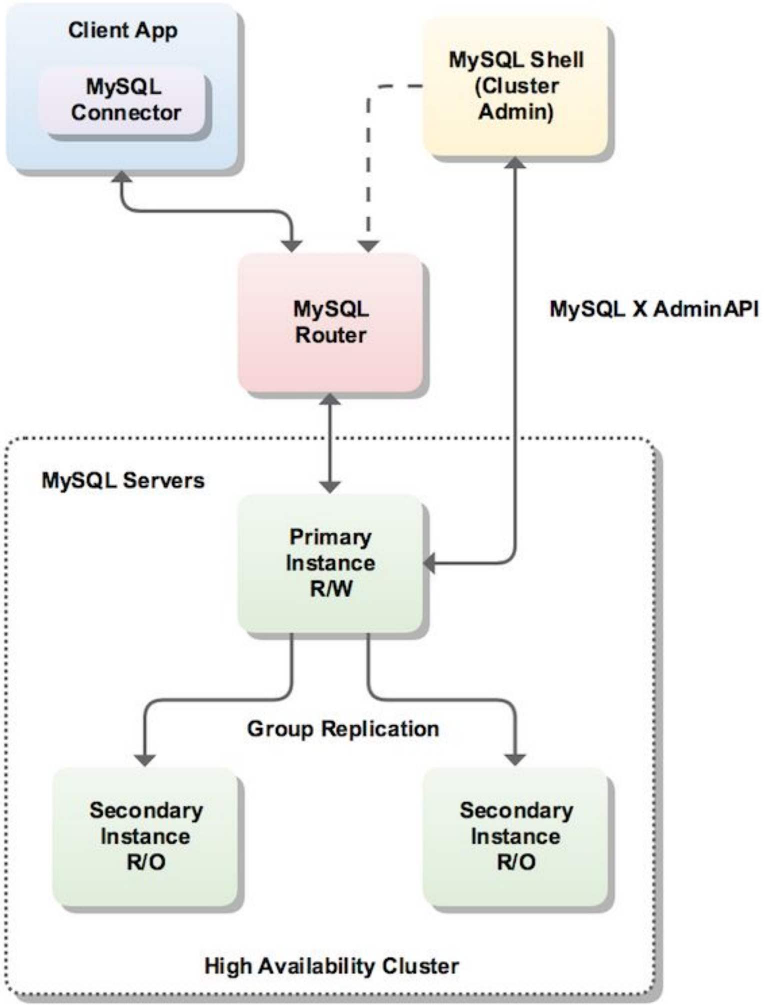

Image from https://dev.mysql.com/doc/refman/5.7/en/mysql-innodb-cluster-introduction.html

MySQL Shell

What is MySQL Shell?

New advanced command-line client for the MySQL Server.

Uses the X DevAPI to communicate with a MySQL Server running the X Plugin.

Supports SQL, JavaScript and Python languages.

Sessions: XSession, Node Session, Classic Session.

Allows Interactive and Batch Code Execution.

MySQL Shell Roles

Configure MySQL instances for Cluster.

Verify MySQL instances are valid for InnoDB Cluster membership.

Create InnoDB Cluster.

Check InnoDB Cluster Status.

Manage InnoDB Cluster.

InnoDB Cluster recovery.

MySQL Router

What is MySQL Router?

Lightweight middleware that provides transparent routing between your application and back-end MySQL Servers.

MySQL Router Roles

Aware of the InnoDB Cluster topology.

Know which MySQL instance is the PRIMARY.

Acts as a proxy to hide the multiple MySQL instances.

Handles failover instead of Client.

MySQL Server

MySQL Server Roles

Provides the MySQL Group Replication mechanism to allow data to be replicated within the cluster, with built-in failover.

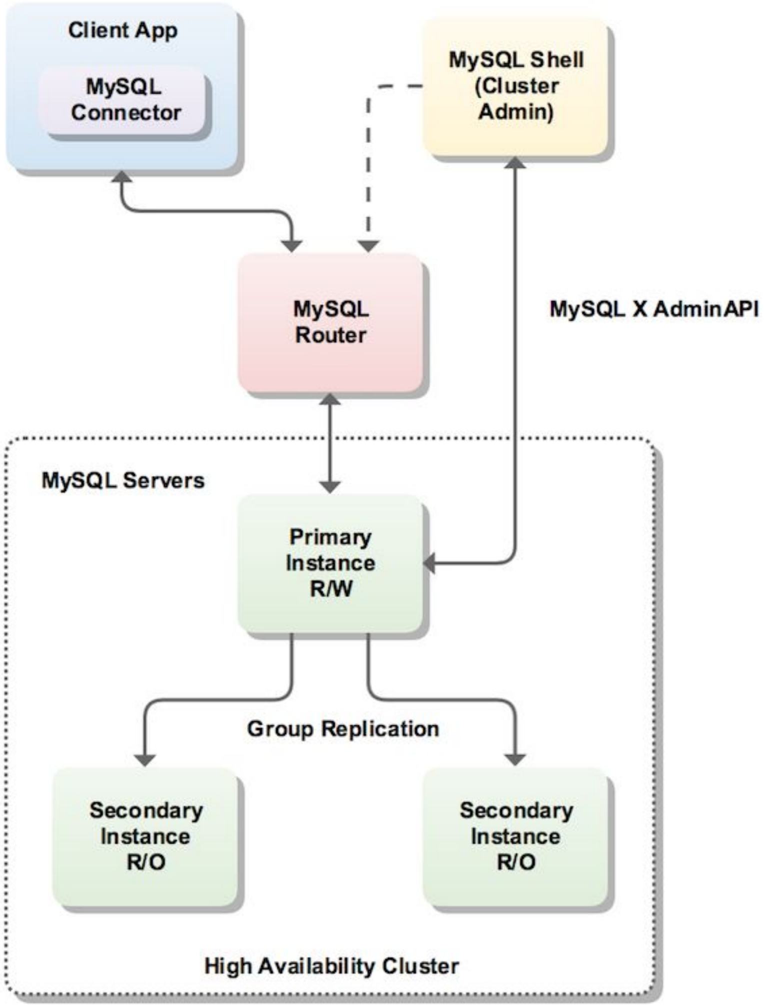

Image from https://dev.mysql.com/doc/refman/5.7/en/mysql-innodb-cluster-introduction.html

More about MySQL InnoDB Cluster

MySQL InnoDB Cluster Modes

Single-primary mode.

Multiple-primaries mode - advanced mode.

Important Limitations and Requirements

At least 3 and not more than 9 servers required for Cluster. Use odd number.

All tables must be of InnoDB type.

All tables must have PRIMARY keys.

Only IPv4 network supported.

Network performance is a must (low latency, high bandwidth).

Supershort How To

Cluster Creation Steps

Set up at least 3 identical MySQL servers, but not more than 9.

Add cluster management user on each server.

Configure MySQL servers with MySQL Shell for Cluster for the first time.

Create MySQL Cluster using MySQL Shell.

Configure MySQL servers with MySQL Shell for Cluster for the second time.

Set up MySQL Routers on the same servers where MySQL Clients are located.

Configure your Clients to use MySQL Routers.

Cluster Creation (internals shortly)

The InnoDB Cluster Metadata Schema is created.

The new InnoDB cluster information, including the specified name and password, is inserted into the InnoDB cluster Metadata.

The seed instance is added to the InnoDB cluster.

The seed instance information is inserted into the InnoDB cluster Metadata.

Connecting to MySQL InnoDB Cluster

If your Clients need to work with InnoDB Cluster, they must be connected to MySQL Router.

Default Read/Write port: 6446 - connects to the PRIMARY.

Default Read/Only port: 6447 - connects to one of the SLAVES.

Only Classic MySQL protocol supported.

The gotchas!

If PRIMARY Node Fails...

If PRIMARY node fails MySQL Router drops all connections from Clients until new PRIMARY elected.

Prepare to handle many “MySQL server has gone away” errors for reads and writes.

Cluster falls apart if all Nodes were down

If all your servers eventually go down in same time your Cluster falls apart and needs to be reconfigured.

This is called “Complete Outage”.

Avoid Large Transactions

Large transactions that cannot be copied between group members over the network within a 5 second window can cause failures in the group communication.

MySQL Router Gotchas

MySQL Router sometimes lags with Cluster updates, especially after recovery.

MySQL Router has weird limit: 500 connections per instance.

MySQL Shell Gotchas

To set up cluster node must be installed on the same host.

During setup must be able to modify MySQL configuration file.

No .deb packages for newer distributions.

Version 1.0.10 has a bug with cluster management user permissions during setup.

No code completion.

Lack of good documentation.

Hangs on lost connection.

Demo!

What We Will Try in the Demo

Restore Cluster from Complete Outage

Kill PRIMARY node during reads.

Restore killed node.

Kill PRIMARY node during writes.

Restore killed node.

What about Production?

I have no experience with MySQL InnoDB Cluster in production.

Some notes for Production

Redundant network.

Redundant power.

Learn to troubleshoot before going live.

Learn how to recover from different failures before going live.

Test failover well before going live.

Big data set might be a problem for initial sync.

Would I run it in Production?

Yes!

But before...

Before going to Production

Before going to Production I would like to try this technology on something small scale and not so vital, as someone's very important and highly loaded project.

Yes, I would need some practice and troubleshooting experience too.

Maybe someone in auditorium already has experience with MySQL InnoDB Cluster and ready to say a word?

Back to the Problem

Would it solve our problem?

Yes, but with additional cost.

You need two more spare servers with MySQL installed.

You need to monitor and maintain MySQL InnoDB Cluster.

Discussion

Sergej Kurakin

Work @mail: sergej.kurakin@nfq.lt

Personal @mail: sergej@kurakin.info

https://www.linkedin.com/in/sergejkurakin