

ice Architectures

ighly available MySQL InnoDB cluster

Infrastructure

Technologies
IaaS, Database, PL/SQL

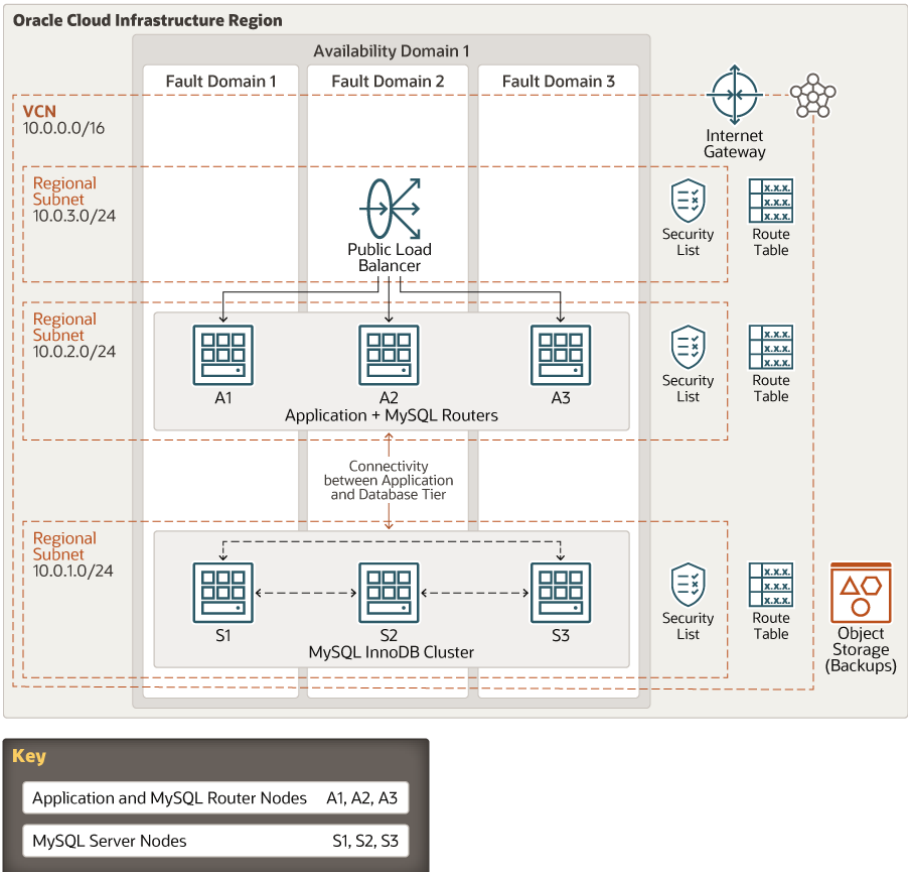
Released
Mar 8, 2022

MySQL is an open source object-relational database management system. It is highly extensible and scalable, and it supports data replication across multiple data centers.

Architecture

This reference architecture shows a typical three-node deployment of a MySQL InnoDB cluster and load-balanced application servers with MySQL routers on Oracle Cloud Infrastructure Compute instances.

The following diagram illustrates this reference architecture.



Description of the illustration [mysql-innodb-cluster-oci.png](#)

[Download diagram](#) [?](#)

The architecture has the following components:

- Region**
An Oracle Cloud Infrastructure region is a localized geographic area that contains one or more data centers, called availability domains. Regions are independent of other regions, and vast distances can separate them (across countries or even continents).
- Availability domains**
Availability domains are standalone, independent data centers within a region. The physical resources in each availability domain are isolated from the resources in the other availability domains, which provides fault tolerance. Availability domains don't share infrastructure such as power or cooling, or the internal availability

Deploy a Highly Available MySQL InnoDB

Architecture

Recommendations

Considerations

Deployment Information

Change Log

[in](#) [t](#) [f](#)

Deploy a Highly Available MySQL InnoDB Cluster

Architecture

Recommendations

Considerations

Deployment Information

Change Log

domain network. So, a failure at one availability domain is unlikely to affect the other availability domains in the region.

- **Fault domains**

A fault domain is a grouping of hardware and infrastructure within an availability domain. Each availability domain has three fault domains with independent power and hardware. When you distribute resources across multiple fault domains, your applications can tolerate physical server failure, system maintenance, and power failures inside a fault domain.

- **Virtual cloud network (VCN) and subnets**

A VCN is a customizable, software-defined network that you set up in an Oracle Cloud Infrastructure region. Like traditional data center networks, VCNs give you complete control over your network environment. A VCN can have multiple non-overlapping CIDR blocks that you can change after you create the VCN. You can segment a VCN into subnets, which can be scoped to a region or to an availability domain. Each subnet consists of a contiguous range of addresses that don't overlap with the other subnets in the VCN. You can change the size of a subnet after creation. A subnet can be public or private.

Every Compute instance is deployed in a VCN that can be segmented into subnets. This architecture uses different public subnets to host the MySQL database servers, application servers, and the public load balancer. If your architecture requires it, you can use private subnets for the application and database servers.

- **Security lists**

For each subnet, you can create security rules that specify the source, destination, and type of traffic that must be allowed in and out of the subnet.

This architecture adds ingress rules for TCP ports 3306, 33060, and 33061 in the security list attached to the MySQL database server subnet. The MySQL server listens on TCP port 3306. TCP port 33060 is used by the X Protocol (from the MySQL router nodes to the database servers). Port 33061 is used for group replication communication.

TCP port 33060 requires an egress rule in the security list attached to the application nodes subnet.

Application-specific TCP ports require ingress rules on the security lists attached to the application server and load balancer subnets, like TCP port 80 for the load balancer.

- **Route table**

Virtual route tables contain rules to route traffic from subnets to destinations outside a VCN, typically through gateways.

- **Internet gateway**

The internet gateway allows traffic between the public subnets in a VCN and the public internet.

- **Load balancer**

The Oracle Cloud Infrastructure Load Balancing service provides automated traffic distribution from a single entry point to multiple servers in the back end.

- **Object storage**

Object storage provides quick access to large amounts of structured and unstructured data of any content type, including database backups, analytic data, and rich content such as images and videos. You can safely and securely store and then retrieve data directly from the internet or from within the cloud platform. You can seamlessly scale storage without experiencing any degradation in performance or service reliability. Use standard storage for "hot" storage that you need to access quickly, immediately, and frequently. Use archive storage for "cold" storage that you retain for long periods of time and seldom or rarely access.

Recommendations

Your requirements might differ from the architecture described here. Use the following recommendations as a starting point.

- **Compute shapes**

This architecture uses an Oracle Linux 7.7 OS image with a VM.Standard2.1 shape for both the application server and the database server. You can choose a different shape if your application's resource needs are different.

- **VCN**

When you create a VCN, determine the number of CIDR blocks required and the size of each block based on the number of resources that you plan to attach to subnets in the VCN. Use CIDR blocks that are within the standard private IP address space.

After you create a VCN, you can change, add, and remove its CIDR blocks.

Use regional subnets.

- **MySQL Server version**

Use MySQL Server version 8.0 for this architecture. We recommend MySQL Enterprise Edition as a best practice.

- **Load balancer bandwidth**

While creating the load balancer, you can either select a predefined shape that provides a fixed bandwidth, or specify a custom (flexible) shape where you set a bandwidth range and let the service scale the bandwidth automatically based on traffic patterns. With either approach, you can change the shape at any time after creating the load balancer.

Considerations

- **Scalability**

This architecture deploys one MySQL database server in each fault domain. It also has one application server with a MySQL router in each fault domain.

You can vertically scale the MySQL InnoDB cluster by changing the virtual machine (VM) shape of each Compute node. Using a shape with a higher core count increases the memory allocated to the Compute instance and network bandwidth.

You can scale out the MySQL InnoDB cluster by adding more nodes to each fault domain.

- **Application availability**

Fault domains provide the best resilience within a single availability domain. This architecture uses another approach to high availability: deploying Compute instances that perform the same tasks across multiple fault domains. This design removes a single point of failure by introducing redundancy.

This architecture uses MySQL InnoDB Cluster on the database tier for its key capabilities like fault tolerance, zero data loss, automatic failover, and active update. If your application can use a simple replication architecture for the database tier and doesn't require automatic failover capabilities, you can use MySQL InnoDB ReplicaSet.

After the architecture is deployed, you can connect to the public IP address of the MySQL node by using SSH tools like PuTTY or Git Bash. You can use the MySQL client for DDL and DML operations on the MySQL database. You can set up the InnoDB cluster and do administration tasks through MySQL Shell.

- **Cost**

Select the VM shape based on the cores, memory, and network bandwidth that you need for your database. You can start with a one-core shape. If you need more performance, memory, or network bandwidth for the application or database node, you can change the VM shape later.

More Information

- [Set up a WordPress CMS connected to a MySQL database in the cloud](#)
- [Deploy Apache Tomcat connected to MySQL Database Service](#)
- [Building Highly Available Applications in a Region with One Availability Domain](#)
- [Best Practices for Deploying High Availability Architecture on Oracle Cloud Infrastructure](#)
- [Learn about InnoDB Cluster](#)

Change Log

This log lists significant changes:

May 12, 2021	Enhanced the content in Architecture and Recommendations.
March 8, 2022	Added downloadable, editable versions of the architecture diagram and links to related reference

	architectures.
--	----------------

Content on Oracle Architecture Center is published for your convenience. It is not part of the Product Documentation for any Oracle product or service.

⊕ Title and Copyright Information

⊕ Documentation Accessibility