

# Oracle® Cloud

## Using Oracle MySQL Cloud Service



18.3  
E75881-22  
July 2018

ORACLE®

Oracle Cloud Using Oracle MySQL Cloud Service, 18.3

E75881-22

Copyright © 2016, 2018, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

# Contents

## Preface

---

Audience	vii
Documentation Accessibility	vii
Related Documents	vii
Conventions	vii

## 1 Getting Started with MySQL Cloud Service

---

About Oracle MySQL Cloud Service	1-1
About MySQL Cloud Service Deployments	1-2
Computing Power	1-2
Database Storage	1-3
Automatic Backup Configuration	1-3
Before You Begin with MySQL Cloud Service	1-4
How to Begin with MySQL Cloud Service Subscriptions	1-5
About MySQL Cloud Service Roles and Users	1-5
Accessing MySQL Cloud Service Console	1-6
Typical Workflow for Using MySQL Cloud Service	1-6
About MySQL Cloud Service Deployments in Oracle Cloud Infrastructure	1-7

## 2 Managing the MySQL Cloud Service Life Cycle

---

Creating a MySQL Server Deployment	2-1
Creating a QuickStart Database Deployment	2-10
Enterprise Edition	2-11
Creating a Cloned Instance from a Snapshot	2-11
Viewing All MySQL Server Deployments	2-12
Viewing Detailed Information for a MySQL Server Deployment	2-13
Viewing Activities for MySQL Server Deployments in an Identity Domain	2-13
Stopping, Starting and Restarting	2-13
Rebooting a Compute Node	2-17
Deleting a MySQL Server Deployment	2-17
Tracking the Number of MySQL Server Deployments in an Account	2-18

Creating and Managing IP Reservations	2-18
Creating an IP Reservation	2-18
Deleting an IP Reservation	2-19
Creating and Managing Snapshots	2-19
Managing Tags	2-21
Creating, Assigning, and Unassigning Tags	2-21
Finding Tags and Instances Using Search Expressions	2-22

### 3 Managing Network Access to MySQL Cloud Service

---

About Network Access to MySQL Cloud Service	3-1
Generating a Secure Shell (SSH) Public/Private Key Pair	3-2
Generating an SSH Key Pair on UNIX and UNIX-Like Platforms Using the ssh-keygen Utility	3-3
Generating an SSH Key Pair on Windows Using the PuTTYgen Program	3-3
Creating an SSH Tunnel to a Compute Node Port	3-4
Creating an SSH Tunnel Using the ssh Utility on Linux	3-5
Creating an SSH Tunnel Using the PuTTY Program on Windows	3-5
Enabling Access to a Compute Node Port	3-7

### 4 Administering MySQL Cloud Service

---

Adding Storage to MySQL Cloud Service	4-1
Scaling MySQL Cloud Service	4-2
Updating or Adding a New SSH Public Key	4-3
Maintaining the Manageability of Your Database Deployments	4-3
Loading Data into the MySQL Database	4-4
Tuning Database Performance on MySQL Cloud Service	4-4

### 5 Accessing MySQL Cloud Service

---

Connecting to a Compute Node Through Secure Shell (SSH)	5-1
Connecting to a Compute Node Using the ssh Utility on UNIX and UNIX-Like Platforms	5-1
Connecting to a Compute Node Using the PuTTY Program on Windows	5-2

### 6 Backing Up and Restoring Databases on MySQL Cloud Service

---

About Backing Up Databases on MySQL Cloud Service	6-1
Creating an On-Demand Backup	6-3
Creating an On-Demand Backup by Using the MySQL Cloud Service Console	6-3
Deleting a Backup	6-4

Customizing the Current Backup Configuration	6-4
Restoring from a Specific Backup	6-5
Restoring from a Specific Backup by Using the MySQL Cloud Service Console	6-5
Restoring to a Specific Point in Time	6-5
Restoring to a Specific Point in Time by Using the MySQL Cloud Service Console	6-6
Updating the Password for Backing Up to the Storage Cloud	6-6

## 7 Patching MySQL Cloud Service

---

About Patching an Oracle MySQL Cloud Service Instance	7-1
Checking Prerequisites Before Applying a Patch	7-2
Checking Patch Prerequisites	7-2
Applying a Patch	7-3
Applying a Patch Using the MySQL Cloud Service Console	7-3
Rolling Back a Patch	7-4
Rolling Back a Patch by Using the MySQL Cloud Service Console	7-4

## 8 Migrating MySQL Databases to MySQL Cloud Service

---

Migrating MySQL Data	8-1
----------------------	-----

## A Characteristics of a Newly Created Deployment

---

Linux User Accounts	A-1
Locations of Installed Software	A-2

## B Oracle Cloud Pages for Administering MySQL Cloud Service

---

MySQL Cloud Service Instances Page	B-1
MySQL Cloud Service Activity Page	B-3
MySQL Cloud Service SSH Access Page	B-4
MySQL Cloud Service IP Reservations Page	B-5
MySQL Cloud Service Overview Page	B-6
MySQL Cloud Service Access Rules Page	B-8
MySQL Cloud Service Backup Page	B-9
MySQL Cloud Service Patching Page	B-10
MySQL Cloud Service Snapshots Page	B-11
Create MySQL Cloud Service: Service Page	B-11
Create MySQL Cloud Service: Service Details Page	B-12
Create MySQL Cloud Service: Supported Timezones	B-16



# Preface

This document describes how to manage and monitor Oracle MySQL Cloud Service and provides references to related documentation.

## Topics

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)

## Audience

This document is intended for Oracle Cloud users who want to manage and monitor Oracle MySQL Cloud Service.

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

### Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

## Related Documents

For more information, see these Oracle resources:

- *Getting Started with Oracle Cloud*
- *Using Oracle Compute Cloud Service (for PaaS)*

## Conventions

The following text conventions are used in this document:

---

<b>Convention</b>	<b>Meaning</b>
<b>boldface</b>	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

---



# 1

## Getting Started with MySQL Cloud Service

This section describes how to get started with Oracle MySQL Cloud Service for administrators and developers.

### Topics

- [About Oracle MySQL Cloud Service](#)
- [About MySQL Cloud Service Deployments](#)
- [Before You Begin with MySQL Cloud Service](#)
- [How to Begin with MySQL Cloud Service Subscriptions](#)
- [About MySQL Cloud Service Roles and Users](#)
- [Accessing MySQL Cloud Service Console](#)
- [Typical Workflow for Using MySQL Cloud Service](#)
- [About MySQL Deployments on OCI](#)

## About Oracle MySQL Cloud Service

Oracle MySQL Cloud Service provides you the ability to deploy MySQL Server in the Cloud, with each deployment containing a single MySQL Server. You have full access to the features and operations available with MySQL Server, with Oracle providing the computing power, physical storage and tooling to simplify maintenance and management operations.

When you create a deployment, MySQL Cloud Service creates compute nodes to host the MySQL Server, using computing and storage resources provided by Oracle Compute Cloud Service. Additionally, it provides access to the compute nodes (and thus to MySQL Server) using networking resources provided by Oracle Compute Cloud Service.

When you create a deployment, MySQL Server and supporting software is installed for you, a MySQL database is created and configured using values you provide when creating the deployment, and the MySQL Server is started. Additionally, you can direct MySQL Cloud Service to set up automatic backups. Finally, the deployment includes cloud tooling that simplifies backup, recovery, and patching operations. You have root privilege, so you can load and run software in the compute environment. You have full administrative privileges for the MySQL Server. You are responsible for making any changes to the automated maintenance setup, and you are responsible for recovery operations in the event of a failure.

MySQL Enterprise Monitor is also delivered as part of the service. This monitoring tool can be installed when the service is created, using the **Create Service** wizard, or from the command line on the compute node.

 **Note:**

If you intend to install MySQL Enterprise Monitor, it is strongly recommended that you dedicate a single MySQL Cloud Service instance to this installation. It is also recommended that this MySQL Server deployed on that instance is not used for any other purpose than the MySQL Enterprise Monitor's repository. A compute node shape of OC5 is the recommended minimum.

### MySQL Server Feature Availability

All MySQL Enterprise Edition Server features are available in Oracle MySQL Cloud Service.

 **Note:**

If MySQL Transparent Data Encryption (TDE) is enabled on the MySQL instance, it is not currently possible to restore a backup of that instance.

## About MySQL Cloud Service Deployments

When you create MySQL Server deployments on MySQL Cloud Service, you use the Create MySQL Cloud Service Instance wizard, which steps you through the process of making the choices that create a deployment tailored to your needs. These choices include:

- [Computing Power](#)
- [Database Storage](#)
- [Automatic Backup Configuration](#)

### Computing Power

When creating a database deployment on MySQL Cloud Service, you choose the computing power for the associated compute node from a list of supported OCPU (Oracle CPU) and processor RAM combinations. The combinations you can choose from depend on the infrastructure you are creating the deployment in:

- In Oracle Cloud Infrastructure:
  - VM.Standard1.1 - 1.0 OCPU, 7.0 GB RAM
  - VM.Standard1.2 - 2.0 OCPU, 14.0 GB RAM
  - VM.Standard1.4 - 4.0 OCPU, 30.0 GB RAM
  - VM.Standard1.8 - 8.0 OCPU, 56.0 GB RAM
  - VM.Standard1.16 - 16.0 OCPU, 240.0 GB RAM
  - VM.Standard2.1 - 1.0 OCPU, 15.0 GB RAM
  - VM.Standard2.2 - 2.0 OCPU, 30.0 GB RAM
  - VM.Standard2.4 - 4.0 OCPU, 60.0 GB RAM
  - VM.Standard2.8 - 8.0 OCPU, 120.0 GB RAM

- BM.Standard1.32 - 36.0 OCPU, 256.0 GB RAM
- BM.Standard2.52 - 52.0 OCPU, 768.0 GB RAM
- In Oracle Cloud Infrastructure Classic:
  - General Purpose:
    - \* OC3 - 1 OCPU, 7.5 GB RAM
    - \* OC4 - 2 OCPU, 15 GB RAM
    - \* OC5 - 4 OCPU, 30 GB RAM
    - \* OC6 - 8 OCPU, 60 GB RAM
    - \* OC7 - 16 OCPU, 120 GB RAM
  - High Memory:
    - \* OC1M - 1 OCPU, 15 GB RAM
    - \* OC2M - 2 OCPU, 30 GB RAM
    - \* OC3M - 4 OCPU, 60 GB RAM
    - \* OC4M - 8 OCPU, 120 GB RAM
    - \* OC5M - 16 OCPU, 240 GB RAM

## Database Storage

When creating a database deployment on MySQL Cloud Service, you choose the amount of usable data storage you want for your database in gigabyte (GB) increments up to a maximum of 1 TB (1024 GB).

## Automatic Backup Configuration

When creating a new database deployment on MySQL Cloud Service, you choose whether you want automatic backups to be configured for the database. Your choices are:

- **Both Cloud Storage and Local Storage**—30 days' worth of backups are kept, with the 7 most recent days' worth available directly on the compute node's local storage. This can be changed later.
- **None**—automatic backups are not configured.

### Note:

Do not use the Oracle Storage Cloud container that you are using to back up MySQL Cloud Service databases to cloud storage for any other purpose. For example, do not also use it to back up Oracle Java Cloud Service instances to cloud storage. Using the container for multiple purposes can result in billing errors.

# Before You Begin with MySQL Cloud Service

Before you begin using MySQL Cloud Service, you should be familiar with the following technologies:

- Oracle Cloud  
See [Getting Started with Oracle Cloud](#) for information about Oracle Cloud.
- Oracle Compute Cloud Service  
MySQL Cloud Service compute nodes use a variety of Oracle Compute Cloud Service resources. See [Using Oracle Compute Cloud Service](#) for information about instances and the disk images, compute shapes, storage volumes, IP reservations, security lists, security rules, and secure shell (SSH) public keys that are associated with them.
- Oracle Storage Cloud Service containers  
MySQL Cloud Service uses MySQL Enterprise Backup to back up to cloud storage which, in turn, uses Oracle Cloud Infrastructure Object Storage Classic containers as repositories for backups to the cloud. Before you can create a container, you must have access to Oracle Cloud Infrastructure Object Storage Classic. See About Oracle Storage Cloud Service in [Using Oracle Cloud Infrastructure Object Storage Classic](#)

 **Note:**

You must enable a Replication Policy on your Oracle Storage Cloud Service before attempting to use it with MySQL Cloud Service. For more information, see [Selecting a Replication Policy for Your Service Instance](#).

Before you create a MySQL Cloud Service instance:

- Procure a MySQL Cloud Service subscription. Without an active subscription, you cannot create a MySQL Cloud Service instance.
- Create a Secure Shell (SSH) public/private key pair. The SSH keys are used to facilitate secure access to the compute nodes that support your database deployments. See [Generating a Secure Shell \(SSH\) Public/Private Key Pair](#). It is also possible to create the key pair during the provisioning tasks.
- (Optional) Create a container in Oracle Storage Cloud Service to store backups on cloud storage. See [Creating Containers](#) in [Using Oracle Storage Cloud Service](#). It is also possible to create a storage container during provisioning. For more information, see [Creating a MySQL Server Deployment](#).

 **Note:**

Do not use an Oracle Storage Cloud container that you use to back up MySQL Cloud Service databases for any other purpose. For example, do not also use it to back up Oracle Java Cloud Service instances. Using the container for multiple purposes can result in billing errors.

## How to Begin with MySQL Cloud Service Subscriptions

Here's how to get started with MySQL Cloud Service trials and paid subscriptions:

1. Request a trial environment or purchase a subscription to Oracle Cloud Services. See *Subscribing to an Oracle Cloud Service Trial, Buying a Metered Subscription to an Oracle Cloud Service, or Buying a Nonmetered Subscription to an Oracle Cloud Service* in *Getting Started with Oracle Cloud*.
2. Set up your Oracle Cloud Services account. See *Setting Up a Metered Oracle Cloud Services Account or Buying a Nonmetered Subscription to an Oracle Cloud Service* in *Getting Started with Oracle Cloud*.
3. Verify MySQL Cloud Service is ready to use. See *Verifying That Metered Oracle Cloud Services Are Running or Verifying That a Service Is Running* in *Getting Started with Oracle Cloud*.
4. Learn about MySQL Cloud Service users and roles. See [About MySQL Cloud Service Roles and Users](#).
5. Create accounts for your users and assign them appropriate privileges and roles. See *Managing User Accounts and Managing User Roles* in *Managing and Monitoring Oracle Cloud*.
6. Be sure to review the prerequisites described in [Before You Begin with MySQL Cloud Service](#) before you create your first MySQL Cloud Service .

Note that an Oracle Public Cloud Services trial environment or purchased subscription comes with Oracle IaaS Public Cloud Services, which provides you access to:

- Oracle Compute Cloud Service, which MySQL Cloud Service relies on
- Oracle Storage Cloud Service, which MySQL Cloud Service relies on for backup storage.

Despite the fact that these services are part of the Oracle IaaS Public Cloud Services subscription, use of their features by MySQL Cloud Service are billed to the Oracle Public Cloud Services subscription.

## About MySQL Cloud Service Roles and Users

In addition to the roles and privileges described in *Oracle Cloud User Roles and Privileges* in *Getting Started with Oracle Cloud*, the following role is created for MySQL Cloud Service: MySQLCS MySQLAdministrators.

When the MySQL Cloud Service account is first set up, the service administrator is given this role. User accounts with this role must be added before anyone else can access and use MySQL Cloud Service.


The identity domain administrator can create more MySQL Cloud Service administrators by creating user accounts and assigning the role to user. For information about how to add user accounts in Oracle Cloud, see *Managing User Accounts* and *Managing and Monitoring Oracle Cloud*.

The following table summarizes the privileges given to a **Tenant User** in MySQL Cloud Service:

Description of Privilege	More Information
Can create and delete MySQL Server deployments	<a href="#">Creating a MySQL Server Deployment</a> <a href="#">Deleting a MySQL Server Deployment</a>
Can scale, patch, and back up or restore MySQL Server deployments	<a href="#">Scaling MySQL Cloud Service</a> <a href="#">Patching MySQL Cloud Service</a> <a href="#">Backing Up and Restoring Databases on MySQL Cloud Service</a>
Can monitor and manage service usage in Oracle Cloud	<a href="#">Managing and Monitoring Oracle Cloud Services in <i>Managing and Monitoring Oracle Cloud</i></a>

## Accessing MySQL Cloud Service Console

To access the MySQL Cloud Service console:

1. Sign in to your Cloud Account and go to the My Services Dashboard.  
See [Signing in to Your Cloud Account](#) in *Getting Started with Oracle Cloud*.
2. Click the  navigation menu in the top corner of the My Services Dashboard and then click **MySQL**.  
The MySQL Cloud Service console opens.
3. If a Welcome page is displayed, go to the Instances page by clicking **Instances**.

For information about the details provided on the MySQL Cloud Service console, see [MySQL Cloud Service Instances Page](#).

## Typical Workflow for Using MySQL Cloud Service

To start using MySQL Cloud Service, refer to the following tasks as a guide:

Task	Description	More Information
Sign up for a free credit promotion or purchase a subscription (Does not apply to Oracle Cloud Machine)	Provide your information, and sign up for a free trial or purchase a subscription to Oracle MySQL Cloud Service	<a href="#">How to Begin with MySQL Cloud Service Subscriptions</a>
Add and manage users and roles	Create accounts for your users and assign them appropriate privileges. Assign the necessary MySQL Cloud Service roles.	<a href="#">Managing User Accounts and Managing User Roles in <i>Managing and Monitoring Oracle Cloud</i></a> , and <a href="#">About MySQL Cloud Service Roles and Users</a>
Create an SSH key pair	Create SSH public/private key pairs to facilitate secure access to the compute nodes associated with your database deployments.	<a href="#">Generating a Secure Shell (SSH) Public/Private Key Pair</a>
Create a MySQL Server deployment	Use a wizard to create a new MySQL Server deployment.	<a href="#">Creating a MySQL Server Deployment</a>

Task	Description	More Information
Enable network access	Permit access to network services associated with your MySQL Server deployments.	<a href="#">About Network Access to MySQL Cloud Service</a>
Load data into the database	Use standard MySQL Server tools to load data into your databases.	<a href="#">Loading Data into the MySQL Database</a>
Monitor database deployments	Check on the health and performance of individual MySQL Server deployments.	<a href="#">MySQL Enterprise Monitor 4.0 Manual</a>
Monitor the service	Check on the day-to-day operation of your service, monitor performance, and review important notifications.	Managing and Monitoring Oracle Cloud Services in <i>Managing and Monitoring Oracle Cloud</i>
Patch a database deployment	Apply a patch or roll back a patch.	<a href="#">Patching MySQL Cloud Service</a>
Back up a database deployment	Back up a database or restore a database from a backup.	<a href="#">Backing Up and Restoring Databases on MySQL Cloud Service</a>

## About MySQL Cloud Service Deployments in Oracle Cloud Infrastructure

You can create Oracle MySQL Cloud Service database deployments in Oracle Cloud Infrastructure and in Oracle Cloud Infrastructure Classic.

The environment that your database deployment provides in either type of infrastructure is substantially the same. A few differences exist in the underlying infrastructure components and in the supported capabilities. Awareness of these differences will help you choose an appropriate infrastructure when creating a database deployment.

- **Prerequisite steps before creating your first database deployment:** In Oracle Cloud Infrastructure, you must perform the steps described in [Prerequisites for Oracle Platform Services on Oracle Cloud Infrastructure](#) in the Oracle Cloud Infrastructure documentation. In Oracle Cloud Infrastructure Classic, you must set the replication policy in Oracle Cloud Infrastructure Object Storage Classic before you create database deployments with backups to cloud storage, as described in [Selecting a Replication Policy for Your Service Instance in Using Oracle Cloud Infrastructure Object Storage Classic](#).
- **Regions and availability domains:** While creating a database deployment, you select a region in Oracle Cloud Infrastructure Classic or Oracle Cloud Infrastructure. If you select a region in Oracle Cloud Infrastructure, then you also select an *Availability Domain*. A region in Oracle Cloud Infrastructure has multiple isolated availability domains, each with separate power and cooling. The availability domains within a region are interconnected using a low-latency network.
- **Subnets and IP networks:** In Oracle Cloud Infrastructure Classic, you can optionally attach a database deployment to an IP network that you define beforehand. In Oracle Cloud Infrastructure, you must attach each database deployment to a subnet, which is a part of a virtual cloud network that you create in Oracle Cloud Infrastructure.

- **Compute shapes:** The range of compute shapes that you can select from when creating a database deployment is different for Oracle Cloud Infrastructure and Oracle Cloud Infrastructure Classic. For more information, see [Computing Power](#).
- **Backups to cloud storage:** In Oracle Cloud Infrastructure, an Oracle Cloud Infrastructure Object Storage *bucket* is used to store backups to cloud storage. You must create a storage bucket before you create database deployments configured to back up to cloud storage. In Oracle Cloud Infrastructure Classic, an Oracle Cloud Infrastructure Object Storage Classic *container* is used to store backups to cloud storage. You can create a storage container before you create database deployments configured to back up to cloud storage, or you can have a storage container created at the same time as a database deployment.
- **IP reservations:** Currently, Oracle Cloud Infrastructure does not support IP reservations.
- **Network access to database deployments:** Regardless of the infrastructure that you create your database deployment in, the rules to provide network access to the deployment are preconfigured for you. The interfaces that you use to manage these rules depend on the infrastructure that the deployment is created in:
  - For deployments in Oracle Cloud Infrastructure, you configure the rules, called *security rules*, in the Oracle Cloud Infrastructure interfaces.
  - For deployments in Oracle Cloud Infrastructure Classic, you configure the rules, called *access rules*, in the Oracle MySQL Cloud Service interfaces. Note that these access rules prohibit access by default (with the exception of SSH access on port 22), and you must enable them to provide access to other ports.
- **Scaling database deployments:** In Oracle Cloud Infrastructure, you cannot scale the shape of a database deployment's compute nodes; you can scale only the storage. The minimum size of a new storage volume in Oracle Cloud Infrastructure is 50 GB.
- **Snapshots and deployments cloned from snapshots:** Currently, Oracle Cloud Infrastructure does not support creating storage snapshots of single-instance database deployments and then creating cloned deployments from the snapshots.



# 2

## Managing the MySQL Cloud Service Life Cycle

This section describes tasks to manage the life cycle of MySQL Cloud Service.

### Topics

- [Creating a MySQL Server Deployment](#)
- [Creating a Quickstart MySQL Cloud Service Deployment](#)
- [Creating a Cloned Instance from a Snapshot](#)
- [Viewing All MySQL Server Deployments](#)
- [Viewing Detailed Information for a MySQL Server Deployment](#)
- [Viewing Activities for MySQL Server Deployments in an Identity Domain](#)
- [Stopping, Starting and Restarting](#)
- [Rebooting a Compute Node](#)
- [Deleting a MySQL Server Deployment](#)
- [Tracking the Number of MySQL Server Deployments in an Account](#)
- [Creating and Managing IP Reservations](#)
- [Creating and Managing Snapshots](#)

## Creating a MySQL Server Deployment

To create a database deployment on MySQL Cloud Service, use the Create MySQL Cloud Service wizard as described in the following procedure.

### Before You Begin

Before you create a MySQL Server deployment on MySQL Cloud Service, ensure you have created the following:

- An SSH public/private key pair

When creating a MySQL Server deployment, you must associate an SSH public key with the compute infrastructure supporting the deployment. You can upload a file containing the public key value or you can paste in the value of a public key.

An SSH public key is used for authentication when you use an SSH client to connect to a compute node associated with the deployment. When you connect, you must provide the private key that matches the public key.

You generate an SSH public/private key pair using a standard SSH key generation tool. For more information, see [Generating a Secure Shell \(SSH\) Public/Private Key Pair](#).

 **Note:**

It is also possible to create an SSH key pair in the **Create MySQL Cloud Service Instance** wizard.

- A cloud storage backup location (Optional)

If you want to automatically back up your database to cloud storage, you must associate it with a cloud storage backup location. The type of location you specify depends on the infrastructure the deployment is built on:

- **Oracle Cloud Infrastructure:** cloud backups are stored in an Oracle Cloud Infrastructure Object Storage bucket. You must create a storage bucket before you create database deployments configured to back up to cloud storage.
- **Oracle Cloud Infrastructure Classic:** cloud backups are stored in an Oracle Cloud Infrastructure Object Storage Classic container. You can create the container beforehand and provide the wizard with information about it, or you can have the wizard create the container for you.

 **Note:**

The **Cloud Storage Container** field of the **Create New Instance** wizard on Oracle Cloud Infrastructure Classic is now populated with a default, valid URL, and a default container name, **MySQLCS**. This container is created the first time you create a new instance. To use a different container name, edit the url to remove the default container name, and replace with the desired name.

The container becomes associated with MySQL Cloud Service. Once associated with MySQL Cloud Service, the container becomes part of your subscription (or trial) rather than part of an Oracle IaaS Public Cloud Services subscription (or trial).

Whether you create the container beforehand or have the wizard do it for you, you are prompted for the following information about the container:

- \* The name of the container
- \* The user name and password of a user who has read/write access to the container

 **Note:**

Do not use the Oracle Cloud Infrastructure Object Storage Classic container that you are using to back up MySQL Cloud Service databases for any other purpose. For example, do not also use it to back up Oracle Java Cloud Service instances. Using the container for multiple purposes can result in billing errors.

## Procedure

To create a MySQL Server deployment on MySQL Cloud Service:

1. Open the MySQL Cloud Service console.

For detailed instructions, see [Accessing MySQL Cloud Service Console](#).

2. Click **Create Instance**.

The Create MySQL Cloud Service wizard starts and the **Create New Instance** page is displayed. The Create New Instance wizard is divided into **Instance**, **Service Details**, and **Confirm** pages

3. On the **Instance** section, specify the following information for your MySQL Server deployment:

- **Instance Name**—enter a name for your MySQL Server deployment. This name:
  - Must not exceed 50 characters.
  - Must start with a letter.
  - Must contain only letters, numbers, or hyphens.
  - Must not contain any other special characters.
  - Must be unique within the identity domain.
- **Description**—enter a description for your MySQL Server deployment. (Optional)
- **Email Address**—provide an email address for updates on the provisioning status.
- **Metering Frequency**—determines the type of metering used to determine billing for resources used by this instance. Currently, only Hourly is supported by MySQL Cloud Service.
- **Region**—name of the Compute Region. (Available only if your identity domain is enabled for regions.)

The region for the database deployment. If you choose a region that supports Oracle Cloud Infrastructure, the **Availability Domain** and **Subnet** fields are displayed, and the deployment will be created on Oracle Cloud Infrastructure. Otherwise, the deployment will be created on Oracle Cloud Infrastructure Classic.

Choose **No Preference** to let MySQL Cloud Service choose an Oracle Cloud Infrastructure Classic region for you. The **Default Data Region** you defined when setting up your account is used.

- **Availability Domain**— (Available only on Oracle Cloud Infrastructure) The availability domain, within the selected region, where your deployment is placed.
- **Subnet**— (Available only on Oracle Cloud Infrastructure) The subnet (within the availability domain) which determine network access to the deployment.
- **IP Network**— (Available only on Oracle Cloud Infrastructure)

(Available only if you selected a region and defined one or more IP networks created in that region using Oracle Cloud Infrastructure Compute Classic.)

Select the IP network where you want the database deployment placed.

Choose **No Preference** to use the default shared network provided by Oracle Cloud Infrastructure Compute Classic.

For more information about IP networks, see these topics in *Using Oracle Cloud Infrastructure Compute Classic*:

- About IP Networks
- Creating an IP Network
- **Tags**—metadata used to identify instances. Select the required tags from the drop-down list, or create the required tags. Tags can be defined as a comma-separated list of **keys**, or as a **key:value** pairs. For example: **productionDB,owner:John,admin:John**. You can search and filter on tags from the **Instances** page. You can also define tags after the instance is created, from the **Instance Overview** page.

For more information, see [Managing Tags](#).

Then, click **Next** to navigate to the **Service Details** section.

 **Note:**

An asterisk adjacent to a field name indicates that the field is required.

- a. In the **Configuration** section, select the name, description, and compute shape of your MySQL Server deployment and provide an SSH public key to access the compute infrastructure supporting the deployment.
  - **Shape**—select a compute shape from the list of supported OCPU (Oracle CPU) and RAM combinations. For more information, see [Computing Power](#).

 **Note:**

It is possible to scale the shape up or down, depending on your requirements. For more information, see [Scaling MySQL Cloud Service](#).

- **SSH Public Key**—provide the SSH public key to be used for authentication when using an SSH client to connect to a compute node that is associated with your MySQL Server deployment.

Click **Edit** to specify the public key, either by uploading a file containing the public key value or by pasting in the value of a public key.

If you paste in the value, make sure the value does not contain line breaks or end with a line break.

Click **Create a New Key** and **Enter** to generate and download a new SSH key pair.
  - **Use High Performance Storage**—select to use high-performance storage for the Data, Redo, and Binlog volumes. This is solid state drive storage.
- b. **Reserved IPs**—select the reserved IP address you want to assign to this service. To assign any of the Reserved IP addresses, select **Assign Automatically**. For more information on reserved IP addresses, see [IP Reservations](#)

This field is displayed only if a Region was selected on the preceding screen, and the **IP Network** field was set to **No Preference**.

- c. In the **MySQL Configuration** section, select the amount of data storage, administrator user, administrator password, database schema name.
  - **Usable Database Storage (GB)**—enter the amount of storage you want for database data (in GB). The default is 25, the maximum is 1024.

 **Note:**

If you do not define local backup storage, the maximum available is 2TB.

- **Administration User**—enter the administration user name. The default value is **root**. This value must:
  - Must be between 2 and 32 characters.
  - Must start with a letter.
  - Must contain only letters, numbers, or this symbol: \_ (underscore).
- **Administration Password** and **Confirm Password**—enter and then reenter a password for administrative access to the MySQL Server.

The password you enter:

  - Must be 8 to 30 characters in length.
  - Must contain at least one lowercase letter
  - Must contain at least one uppercase letter
  - Must contain at least one number
  - Must contain at least one of these symbols: \_ (underscore), # (hash sign), or \$ (dollar sign).
- **Database Schema Name**—enter a name for the database instance(s). This name:
  - Must not exceed 64 characters. Can contain letters, numbers and special characters (\_, \$).
  - Must start with a letter, underscore (\_), or dollar sign (\$).
  - Must contain only letters, numbers, or these symbols: \_ (underscore), # (hash sign), or \$ (dollar sign).
- **Server Character Set**—select the required character set from the drop-down menu.

 **Note:**

Although the default character set for MySQL is latin1, MySQL Cloud Service uses utf8mb4 by default.

- **Timezone**—select the timezone for the MySQL server. The default value, **SYSTEM**, sets the timezone to that used by the compute node. See [Supported Timezones](#) for a complete list.

- **MySQL Port**—define the port on which the server listens. The range of available ports is 3200 to 3399. The default is 3306.
- d. **Configure Enterprise Monitor**—select **Yes** to configure a MySQL Enterprise Monitor installation on this instance. Select **No** to carry on without installing MySQL Enterprise Monitor.

 **Note:**

If you intend to install MySQL Enterprise Monitor, it is strongly recommended that you dedicate a single MySQL Cloud Service instance to this installation. It is also recommended that this MySQL Server deployed on that instance is not used for any other purpose than the MySQL Enterprise Monitor's repository. A compute node shape of OC3 is the recommended minimum.

 **Note:**

If MySQL Enterprise Monitor is provisioned, you must also enable the default access rule, **ora\_p2admin\_em**. If this access rule is not enabled, it is not possible to access MySQL Enterprise Monitor.

If you select Yes, the following fields are displayed:

- **Manager User**—enter the name of the MySQL Enterprise Monitor's Manager user. This is the user who configures the installation, adds users, and so on.
- **Manager Password**—enter and confirm a password for the Manager User.

The password you enter:

- Must be 8 to 30 characters in length.
  - Must contain at least one lowercase letter
  - Must contain at least one uppercase letter
  - Must contain at least one number
  - Must contain at least one of these symbols: \_ (underscore), # (hash sign), or \$ (dollar sign).
- **Agent User**—enter the name of the Agent User. This user is used by the MySQL Enterprise Agents to connect to the MySQL Service Manager and relay the information collected about the host on which the agent is installed and any MySQL instances installed on that host.

 **Note:**

No agent installations are provided at this time. It is currently only possible to install the MySQL Enterprise Service Manager, which can monitor any MySQL instances, local or remote, using a built-in agent. If you want to monitor MySQL instances installed on other MySQL Cloud Service instances, you can configure the MySQL Service Manager to monitor them, but cannot monitor a remote host's operating system, file system, or network interfaces. Although agent installations are not currently possible, it is mandatory to define an Agent User and Password to properly configure the MySQL Enterprise Monitor. These parameters can be changed later.

- **Agent Password**—enter, and confirm, the agent password.
- e. In the **Backup and Recovery Configuration** section, choose a backup option for the database deployment and, depending on your choice, provide information about the Oracle Storage Cloud Service container where cloud backups are to be stored.

 **Note:**

It is not possible to change the backup location after the service is created.

**Backup Destination**—select how backups are to be configured:

- **Both Cloud Storage and Local Storage**—backups are configured to be created automatically and stored both on local storage and on an Oracle Storage Cloud Service container.
- **None**—backups are not configured.
- **Cloud Storage Only**—backups are configured to be created automatically and stored in an Oracle Storage Cloud Service container. No local backup is created for this option.

If you select **Cloud Storage Only** or **Both Cloud Storage and Local Storage**, the Cloud Storage Container, User Name and Password fields are displayed:

- **Cloud Storage Container**—The URL where backups to cloud storage are to be stored:
  - For database deployments in Oracle Cloud Infrastructure, enter the URL of an existing Oracle Cloud Infrastructure Object Storage bucket. The URL is of the form:

```
https://swiftobjectstorage.region.oraclecloud.com/v1/namespace/bucket
```

For example:

```
https://swiftobjectstorage.us-phoenix-1.oraclecloud.com/v1/mycompany/mybucket
```

You must create this storage bucket before you begin creating the database deployment. See [Object Storage API](#) in Oracle Cloud Infrastructure documentation.

- For database deployments in Oracle Cloud Infrastructure Classic, the field is populated by default with the URL of an Oracle Cloud Infrastructure Object Storage Classic container. The URL is of the general form:

```
https://storage-instance-specifier/container
```

 **Note:**

If the account does not include an Object Storage service entitlement or if the region selected is an Oracle Cloud Infrastructure region, the container field is not autopopulated.

To use a container other than the default, **MySQLCS**, replace the default container name with the desired name. The new name can be an existing container, or you can create a new container by adding the new name to the URL, and selecting **Create Cloud Storage Container**.

 **Note:**

Do not use the storage container that you are using to back up MySQL Cloud Service deployments to cloud storage for any other purpose. For example, do not also use it to back up Oracle Java Cloud Service instances to cloud storage. Using the container for multiple purposes can result in billing errors.

To determine the *storage-instance-specifier* value for your account, see About REST URLs for Oracle Storage Cloud Service Resources in *Using Oracle Cloud Infrastructure Object Storage Classic*. If this container doesn't exist, use the **Create Cloud Storage Container** checkbox to create it.

- **Cloud Storage User Name**—A user with read/write (and creation, if necessary) access to the location specified in **Cloud Storage Container**:
  - For database deployments in Oracle Cloud Infrastructure, enter the user name you use to sign in to the Oracle Cloud Infrastructure console.
  - For database deployments in Oracle Cloud Infrastructure Classic, enter the Oracle Cloud user name of the administrator of the Oracle Cloud Infrastructure Object Storage Classic container specified in **Cloud Storage Container**. Usually, this is your Oracle Cloud user name.
- **Cloud Storage Password**—The password necessary to access the location specified in **Cloud Storage Container**



- For database deployments in Oracle Cloud Infrastructure, enter the Auth Token generated in Oracle Cloud Infrastructure for the user that you specified. For more information, see [Prerequisites for Oracle Platform Services on Oracle Cloud Infrastructure](#) in the Oracle Cloud Infrastructure documentation..
  - For database deployments in Oracle Cloud Infrastructure Classic, enter the password of the Oracle Cloud user specified in **Cloud Storage User Name**.
  - **Create Cloud Storage Container**— (Not available on Oracle Cloud Infrastructure) Create a new Oracle Cloud Infrastructure Object Storage Classic container as part of the database deployment creation. Specify the container name and the Cloud Storage user name and password in the preceding fields. Enabled by default.
4. To populate, or initialize, the new database with the data stored in an existing backup, in cloud storage, select **Yes** in the **Create Instance from Existing Backup** field and do the following:

 **Note:**

This process supports MySQL Enterprise Backup Full or TTS, compressed, single-file backups, with the .mbi extension, or mysqldump's logical backups, gzipped (\*.sql.gz).

- **Backup File Cloud Storage Path**—enter the path to the backup file. The following path types are permitted:
  - For deployments in Oracle Cloud Infrastructure, enter the URL of an existing Oracle Cloud Infrastructure Object Storage bucket. The URL is of the form:
 

```
https://swiftobjectstorage.region.oraclecloud.com/v1/namespace/bucket/backupFileName
```

For example:

```
https://swiftobjectstorage.us-phoenix-1.oraclecloud.com/v1/mycompany/mybucket
```

You must create this storage bucket before you begin creating the database deployment. See [Object Storage API](#) in Oracle Cloud Infrastructure documentation.
  - For database deployments in Oracle Cloud Infrastructure Classic, the URL is of the general form:
 

```
https://storage-instance-specifier/container/backupFileName
```

To determine the *storage-instance-specifier* value for your account, see About REST URLs for Oracle Storage Cloud Service Resources in *Using Oracle Cloud Infrastructure Object Storage Classic*.
- **Username**—username of the user with read access to the container specified.
- **Password**—password of the user specified in **Username**.

 **Note:**

You must define adequate storage space in the **Usable Database Storage** field. If the usable storage is less than the size of the backup used, an error is displayed.

5. On the Confirmation page, review the information listed. If you are satisfied with the information, click **Create**.

If you need to change the information, use the navigation bar or **Back** button at the top of the wizard to step back through the pages in the wizard. Click **Cancel** to cancel out of the wizard without creating a new database deployment.

## Creating a QuickStart Database Deployment



This topic does not apply to Oracle Cloud Infrastructure.

For accounts that use the Universal Credits payment model, MySQL Cloud Service provides a QuickStart template that creates a deployment of commonly used configuration. You simply pick a template and, if desired, change the default deployment name. MySQL Cloud Service then uses an Oracle Cloud Stack Manager template to provide all the other configuration information.

### Procedure

1. Go to the QuickStarts page:
  - a. Open the MySQL Cloud Service console.
  - b. Click the **QuickStarts** link.

The QuickStarts page is displayed.

2. If desired, change the **Instance Name** from the default provided. The name must start with a letter, and can contain up to 25 letters or numbers.
3. Select the template you want to use.

For more information about the database deployments these templates create, see [Enterprise Edition Template](#).

 **Note:**

Currently, MySQL Cloud Service provides only one template, Enterprise Edition.

4. Click the **Create** button below the template you want to use.

The Confirmation window is displayed. Note that the **Create** button is disabled.

 **Note:**

If you want to create a custom service instead of using a template, click the **Custom** button to open the **Create Service** wizard. For more information, see [Creating a MySQL Server Deployment](#).

5. Click the **Download** link and save the zip file containing the SSH key pair and administrator password that will be used to create your deployment. You will need this information to access your database deployment after it is created.

The **Create** button is now enabled.

6. Click **Create**.

The Confirmation window closes, MySQL Cloud Service begins creating the deployment using the Oracle Cloud Stack Manager template, and the Instances page is displayed.

7. Periodically refresh the Instances page to monitor the creation of your new deployment.

## Enterprise Edition

When you use the Enterprise Edition Quickstart option, MySQL Cloud Service uses an Oracle Cloud Stack Manager template to create a deployment consisting of a MySQL database housed on a single compute node. As such, the database deployment is associated with an Oracle Cloud Stack of the same name. In brief, the template performs these actions:

- Creates a compute node of 1 oc3 OCPU with 7.5 GB RAM and its own public IP address.
- Deploys MySQL Server Enterprise Edition on the compute node.
- Creates 125GB of storage and allocates 25GB for the database.
- Creates Oracle Cloud Infrastructure Compute Classic networking resources to provide access to the compute node, setting all except SSH access on port 22 to a disabled status.
- Creates a MySQL database schema named **mydatabase**.
- Starts the database.

Note that the template **does not** configure backups for the deployment, nor is MySQL Enterprise Monitor configured.

## Creating a Cloned Instance from a Snapshot



This topic does not apply to Oracle Cloud Infrastructure.

You can create a MySQL Cloud Service database deployment from a snapshot you have taken of another database deployment in the same identity domain. The resulting

deployment is known as a linked clone because its storage is linked to the snapshot's storage.

### Procedure

1. Open the MySQL Cloud Service console.
2. Click the name of the service whose snapshot you want to use as the basis for a linked clone deployment.  
The MySQL Cloud Service Overview page is displayed.
3. Click the Administration tile and then click the Snapshots tab.  
The MySQL Cloud Service Snapshots page is displayed.
4. In the Available Storage Snapshots list, locate the snapshot you want to create a linked-clone deployment from and choose **Create Clone** from that snapshot's action menu.  
The Create Service wizard is displayed.
5. Step through the pages of the wizard to provide information about the linked-clone deployment.  
Several options are not selectable; for example, Database Schema Name or Server Character Set. Such options are not selectable because their values are determined from the snapshot upon which the linked-clone is based.  
Also note that some options are required: you must provide a new service name, specify an SSH public key, and provide an administrator password. You can change the other selectable from their defaults if you want to; for example, Shape and Backup Destination.
6. After completing the wizard by clicking **Create** on the Confirmation page, the Instances page is displayed, including notice that creation of the database deployment has begun.

## Viewing All MySQL Server Deployments

From the MySQL Cloud Service Console, you can:

- View the total resources allocated across all MySQL Cloud Service instances.
- View the details for each deployment.
- Use the search field to filter the list to include only the deployments that contain a given string in their name.

To view all MySQL Server deployments:

1. Open the MySQL Cloud Service console.

For detailed instructions, see [Accessing MySQL Cloud Service Console](#).

The MySQL Cloud Service console is displayed, showing a list of all MySQL Server instances.

For information about the details provided on the MySQL Cloud Service console, see [MySQL Cloud Service Instances Page](#).

## Viewing Detailed Information for a MySQL Server Deployment

From the MySQL Cloud Service Overview page, you can:

- View a summary of details for a database deployment on MySQL Cloud Service, such as description, subscription mode, and so on.
- View the total resources allocated to the deployment.
- View the details and status information for each node associated with the deployment.

To view detailed information for a MySQL Server deployment:

1. Open the MySQL Cloud Service console.  
For detailed instructions, see [Accessing MySQL Cloud Service Console](#).
2. Select the MySQL Server deployment for which you want to view more information.

The MySQL Cloud Service Overview page is displayed .

For more information about the details provided on this page, see [MySQL Cloud Service Overview Page](#).

## Viewing Activities for MySQL Server Deployments in an Identity Domain

Use the **Activity** page to view activities for MySQL Server deployments on MySQL Cloud Service in your identity domain. You can restrict the list of activities displayed using search filters.

To view activities for your MySQL Server deployments:

1. Open the MySQL Cloud Service console.  
For detailed instructions, see [Accessing MySQL Cloud Service Console](#).
2. Click the **Activity** item.  
The Activity page is displayed, showing the list of all activities started within the past 24 hours. You can use the Start Time Range field to specify a start time range other than the default of the previous 24 hours.
3. Use the options in the Search Activity Log section to filter the results to meet your needs. You can search on start time range, full or partial service name, activity status, and operation type. Click **Search**. View the results in the table that follows.

For information about features on this page, see [MySQL Cloud Service Activity Page](#).

## Stopping, Starting and Restarting

From the MySQL Cloud Service console, you can stop, start and restart the compute nodes associated with a MySQL Server deployment on MySQL Cloud Service.

### Topics

- [About Stopping, Starting and Restarting a MySQL Server Deployment](#)
- [Stopping a MySQL Server Deployment](#)
- [Starting a Stopped MySQL Server Deployment](#)
- [Restarting a MySQL Server Deployment](#)
- [Viewing Past Stop, Start and Restart Activity](#)

## **About Stopping, Starting and Restarting a MySQL Server Deployment**

### **About Stopping a Database Deployment**

When you stop MySQL Cloud Service, no access is possible and you cannot perform management operations on it except to start it or to delete it.

Stopping a MySQL Server deployment is similar to turning off your personal computer; it has no computing capabilities because the CPU and RAM have no power, but all its other resources—disk drives and the data they contain, static IP reservations, and so on—remain and are ready to be put back into use when power is restored.

When MySQL Server deployment is stopped, its CPU and RAM (an Oracle Compute Cloud Service instance) are stopped. As a consequence, it consumes no OCPU or memory resources and so metering and billing of these resources stop. However, all the other resources of the database deployment continue to exist and so continue to be metered and billed, including:

- Oracle Compute Cloud Service resources such as storage volumes and IP address reservations
- Oracle Storage Cloud Service storage space used by the database deployment's backups to the Oracle Cloud (if the MySQL Server deployment was being backed up to cloud storage)

Additionally, when MySQL Server deployment is stopped, backups are not performed.

### **About Starting a Stopped MySQL Server Deployment**

When you start a stopped MySQL Cloud Service database deployment, access becomes possible again and you can perform management operations on it such as scaling and patching.

Starting a stopped database deployment is similar to turning your personal computer back on: its computing capabilities are restored because the CPU and RAM again have power, and all its other resources are put back into use.

When a database deployment is started:

1. An Oracle Compute Cloud Service instance of the appropriate compute shape (OCPU and memory) is allocated to it.
2. All other Compute Cloud Service resources associated with it when it was created or as the result of a scaling operation are reattached to it.
3. The allocated Oracle Compute Cloud Service instance is started.

After these steps complete, the MySQL Server deployment is running and available.

Because the MySQL Server deployment again consumes OCPU and memory resources, metering and billing of these resources resume.

### **About Restarting a MySQL Server Deployment**

When you restart a MySQL Cloud Service database deployment, it is stopped and then immediately started again. Thus, the information about what happens when stopping and starting a database deployment applies to restarting a database deployment as well, just in immediate succession.

### Stopping a MySQL Server Deployment

In general, you stop a MySQL Cloud Service database deployment for one of these reasons:


- To prohibit access to it.
- To reduce its cost of operation, especially if its billing frequency is hourly.

#### Before You Begin

To learn what happens when you stop a MySQL Cloud Service database deployment, review [About Stopping, Starting and Restarting a MySQL Server Deployment](#).

#### Procedure

To stop a database deployment:

1. Open the MySQL Cloud Service console.  
For detailed instructions, see [Accessing MySQL Cloud Service Console](#).
2. In the list of deployments, click the name of the database deployment you want to stop.  
The MySQL Cloud Service Overview page is displayed.
3. From the  menu for the database deployment's compute node, select **Stop**, and then confirm the action.

The deployment first has a status of **Maintenance** and then **Stopped** in the MySQL Cloud Service console.



#### Note:

It is not possible to scale a stopped deployment.

### Starting a Stopped MySQL Server Deployment

#### Before You Begin


To learn what happens when you start a stopped MySQL Cloud Service deployment, review [About Stopping, Starting and Restarting a MySQL Server Deployment](#).

#### Procedure

To start a stopped MySQL Server deployment:

1. Open the MySQL Cloud Service console.  
For detailed instructions, see [Accessing MySQL Cloud Service Console](#).
2. In the list of deployments, click the name of the stopped MySQL Server deployment you want to start.

The MySQL Cloud Service Overview page is displayed.

3. From the  menu for the MySQL Server deployment's compute node, select **Start**, and then confirm the action.

The deployment has a status of **Maintenance** in the MySQL Cloud Service console until it is fully started.

### Restarting a MySQL Server Deployment

#### Note:


Restarting a MySQL Server deployment is different from rebooting a compute node of a MySQL Server deployment. Rebooting a compute node, as described in [Rebooting a Compute Node](#), does not restart the MySQL Server deployment. It simply reboots the compute node.

### Before You Begin

To learn what happens when you restart a MySQL Cloud Service deployment, review [About Stopping, Starting and Restarting a MySQL Server Deployment](#).

### Procedure

To restart a MySQL Server deployment:

1. Open the MySQL Cloud Service console.  
For detailed instructions, see [Accessing MySQL Cloud Service Console](#).
2. In the list of deployments, click the name of the MySQL Server deployment you want to restart.  
The MySQL Cloud Service Overview page is displayed.
3. From the  menu for the MySQL Server deployment's compute node, select **Restart**, and then confirm the action.  
The deployment has a status of **Maintenance** in the MySQL Cloud Service console until it is fully restarted.

### Viewing Past Stop, Start and Restart Activity

You can see information about past stop, start and restart activity by viewing the activity log:

1. Open the MySQL Cloud Service console.  
For detailed instructions, see [Accessing MySQL Cloud Service Console](#).
2. In the list, click the name of the MySQL Server deployment whose past activity you want to view.  
The MySQL Cloud Service Overview page is displayed.
3. Click the triangle icon to the left of the Activity title to expand the activity log.  
The activity log shows information about past operations performed on the MySQL Server deployment, with the most recent activity first.
4. Click the icon next to a particular operation to see details about that operation.  
If an operation failed, the details include information about why it failed.



## Rebooting a Compute Node

On occasion, you might find it necessary to reboot a compute node associated with MySQL Cloud Service. Follow these steps to perform the operation.

### Note:

Rebooting a compute node is different from restarting a compute node. Restarting a compute node, as described in [Restarting a MySQL Server Deployment](#), stops and removes the Compute Cloud Service instance on which the compute node is running and then creates and starts a new Compute Cloud Service instance for the compute node. Rebooting a compute node uses a Linux command to restart the Compute Cloud Service instance on which the compute node is running.

1. Connect as the `opc` user to the compute node.  
For detailed instructions, see [Connecting to a Compute Node Through Secure Shell \(SSH\)](#).

2. Start a root-user command shell:

```
$ sudo -s  
#
```

3. Enter the command to reboot the compute node:

```
# reboot
```

### Caution:

Do not use the `halt`, `shutdown` or `shutdown -h` commands to shut down the compute node. Doing so will stop the compute node indefinitely and will require manual intervention by Oracle Cloud system administrators to restart the compute node.

Your connection to the compute node is closed and the compute node reboots.

## Deleting a MySQL Server Deployment

When you no longer require a MySQL Cloud Service, you can delete it.

### Note:

Once deleted, it is not possible to recover the deployment. Any cloud backups are not deleted automatically.

If the service is a source for cloned services, the clones must be deleted before you attempt to delete the source.

To delete a deployment:

1. Open the MySQL Cloud Service console.

For detailed instructions, see [Accessing MySQL Cloud Service Console](#).

2. Locate the deployment you want to delete and select **Delete** from the  menu.

You are prompted to confirm the deletion. Once deleted, the entry is removed from the list of deployments displayed on the MySQL Cloud Service.

## Tracking the Number of MySQL Server Deployments in an Account

You can track the number of database deployments on MySQL Cloud Service across all identity domains using the My Account Dashboard page. Note that the My Account Dashboard page is different from the My Services Dashboard page: My Account Dashboard shows information for your entire account, while My Services Dashboard shows information limited to one identity domain in your account.

To open the My Account Dashboard page, sign in to My Account. By default, the Dashboard page is in focus. You can also click **Dashboard** at any time to display the page.

Click the service name to go to the details page, which displays status history, availability history, usage metrics, and additional information for the selected service.

For information about the details provided on the Dashboard page, see Exploring the My Account Dashboard Page in *Getting Started with Oracle Cloud*.

## Creating and Managing IP Reservations



This topic does not apply to Oracle Cloud Infrastructure.

If your identity domain supports the regional compute infrastructure, you can create and manage IP reservations. You use these reservations when creating MySQL Cloud Service deployments instead of having MySQL Cloud Service assign one for you.

Here are the tasks for creating and managing IP reservations:

- [Creating an IP Reservation](#)
- [Deleting an IP Reservation](#)

### Creating an IP Reservation

If your identity domain supports the regional compute infrastructure, you can create IP reservations for later use when creating MySQL Cloud Service deployments.

1. Go to the IP Reservations Page.
  - a. Open the MySQL Cloud Service console.

- b. Click **IP Reservations**.  
The IP Reservations (for MySQL Cloud Service) page is displayed.
  2. Click **Create**.  
The Create IP Reservation window displays.
  3. Enter a name for the IP reservation and choose the region the reservation will be available from.
  4. Click **OK**.

## Deleting an IP Reservation

When you no longer require an unused IP reservation, you can delete it.

1. Go to the IP Reservations page.
  - a. Open the MySQL Cloud Service console.
  - b. Click **IP Reservations**.  
The IP Reservations (for MySQL Cloud Service) page is displayed.
2. Locate the unused IP reservation you want to delete and then click the X for that reservation.  
You are prompted to confirm the deletion.
3. Click **OK** to confirm deletion of the IP reservation.

## Creating and Managing Snapshots



This topic does not apply to Oracle Cloud Infrastructure.

MySQL Cloud Service supports the creation of snapshots, which you can then use to create new database deployments called linked clones.

When you create a storage snapshot, the database deployment is put into maintenance status and a snapshot of all the storage for the deployment is taken. Then, when you create a linked clone deployment, MySQL Cloud Service creates a new database deployment whose storage volumes are from the snapshot.

Using the “copy on write” technology that Oracle Compute Cloud Service supports for storage volume snapshots, the file data on the linked clone deployment can change without changing the snapshot itself. Thus, you can create several linked clones from the same snapshot to use for application testing or branched application development work.

The following are the tasks for creating and managing snapshots:

- [Creating a Snapshot](#)
- [Listing Cloned Services Created from a Snapshot](#)
- [Deleting a Cloned Service](#)
- [Deleting a Snapshot](#)

### Creating a Snapshot

1. Go to the Snapshots page of the deployment you want to create a snapshot of:
  - a. Open the MySQL Cloud Service console.
  - b. Click the name of the deployment you want to create a snapshot of.  
The MySQL Cloud Service **Overview** page is displayed.
  - c. Select the Administration tile, then click the Snapshots tab.  
The Snapshots page is displayed. Any snapshots already created for the deployment are shown in the Available Storage Snapshots list.
2. Click **Create**. In the Create Storage Snapshot window that is displayed, enter a name (and, optionally, a description) for the snapshot and then click **Create**.  
In the next window that is displayed, confirm that you want to put the database deployment into maintenance mode and create the snapshot by clicking **Create**.

### Listing Cloned Services Created from a Snapshot

1. Open the MySQL Cloud Service console.
2. Click the name of the deployment whose snapshot you want to see linked clone deployments of.  
The MySQL Cloud Service Overview page is displayed.
3. Click the Administration tile and then click the Snapshots tab.  
The MySQL Cloud Service Snapshots page is displayed.
4. In the Available Snapshots list, locate the snapshot you are interested in and check the value displayed next to Linked Clones.  
If the value is zero, no linked clone deployments have been created from the snapshot. Otherwise, click expand to view the list of cloned services.  
When viewing the list of clones, you can click the name of a cloned service to go directly to that service's Overview page.

### Deleting a Cloned Service

1. Open the MySQL Cloud Service console.
2. Locate the snapshot clone you want to delete and choose **Delete** from that snapshot clone's action menu.
3. In the Delete Service window, confirm that you want to delete the clone database deployment by clicking **Delete**.  
Once deleted, the associated entry is removed from the list of database deployments displayed in the MySQL Cloud Service console.

## Deleting a Snapshot

### Note:

You cannot delete a snapshot that has cloned services created from it. You must first delete the associated clone service, as described in [Deleting a Cloned Service](#).

1. Open the MySQL Cloud Service console.
2. Click the name of the deployment whose snapshot you want to delete.  
The MySQL Cloud Service Overview page is displayed.
3. Click the Administration tile and then click the Snapshots tab.  
The MySQL Cloud Service Snapshots page is displayed.
4. In the Available Storage Snapshots list, locate the snapshot you want to delete and choose **Delete** from that snapshot's action menu.
5. In the Delete Storage Snapshot window, confirm that you want to delete the snapshot by clicking **Delete**.

If the window warns you that you cannot delete the snapshot because there are database deployments cloned from it, click **Close** and then delete the linked clone deployments before trying to delete the snapshot.

## Managing Tags

A tag is an arbitrary key or a key-value pair that you can create and assign to your instances. You can use tags to organize and categorize your instances, and to search for them.

### Topics:

- [Creating, Assigning, and Unassigning Tags](#)
- [Finding Tags and Instances Using Search Expressions](#)

## Creating, Assigning, and Unassigning Tags

You can create and assign tags to instances while creating the instances or later. When you no longer need certain tags for an instance, you can unassign them.

To assign tags to an instance or to unassign tags:

1. Navigate to the Overview page for the instance for which you want to assign or unassign tags.
2. This step depends on whether any tags are already assigned to the instance:  
If at least one tag is assigned to the instance, the Overview page shows a **Tags** field.
  - a. Hover over any of the tags in the **Tags** field, until a **More** link is displayed.

- b. Click the **More** link.

If you don't see the **Tags** field, then no tags are currently assigned to the instance.

- a. Click **Manage this service**  in the instance name bar at the top.
- b. Select **Add Tags**.

The Manage Tags dialog box is displayed.

3. Assign or unassign tags, as required:
  - In the **Assign** section, from the **Tags** field, select the tags that you want to assign to the instance.
  - If the tags that you want to assign don't exist, select **Create and Assign** in the **Tags** field, and then enter the required tags in the **Enter New Tags** field.
  - To unassign a tag, in the **Unassign** section, look for the tag that you want to unassign, and click the **X** button next to the tag.
  - To exit without changing any tag assignments for the instance, click **Cancel**.
4. After assigning and unassigning tags, click **OK** for the tag assignments to take effect.

## Finding Tags and Instances Using Search Expressions

A tag is an arbitrary key or a key-value pair that you can create and assign to your instances. You can use tags to organize and categorize your instances, and to search for them. Over time, you might create dozens of tags, and you might assign one or more tags to several of your instances. To search for specific tags and to find instances that are assigned specific tags, you can use filtering expressions.

For example, on the home page of the web console, you can search for the instances that are assigned a tag with the key `env` and any value starting with `dev` (example: `env:dev1`, `env:dev2`), by entering the *search expression* `'env': 'dev%'` in the **Search** field.



Similarly, when you use the REST API to find tags or to find instances that are assigned specific tags, you can filter the results by appending the optional `tagFilter=expression` query parameter to the REST endpoint URL.

- To find specific tags: `GET paas/api/v1.1/tags/{identity_domain}/tags?tagFilter={expression}`
- To get a list of instances that are assigned specific tags: `GET paas/api/v1.1/instancemgmt/{identity_domain}/instances?tagFilter={expression}`

### Syntax and Rules for Building Tag-Search Expressions

- When using cURL to send tag-search API requests, enclose the URL in double quotation marks.

**Example:**

```
curl -s -u username:password -H "X-ID-TENANT-NAME:acme" "restEndpointURL/
paas/api/v1.1/instancemgmt/acme/instances?tagFilter='env'"
```

This request returns all the tags that have the key `env`.

- Enclose each key and each value in single quotation marks. And use a colon (`:`) to indicate a key:value pair.

#### Examples:

```
'env'
'env': 'dev'
```

- You can include keys or key:value pairs in a tag-filtering expression.

Sample Expression	Description	Sample Search Result
'env'	Finds the tags with the key <code>env</code> , or the instances that are assigned the tags with that key.	The following tags, or the instances that are assigned any of these tags:  env:dev env:qa
'env': 'dev'	Finds the tag with the key <code>env</code> and the value <code>dev</code> , or the instances that are assigned that tag.	The following tag, or the instances that are assigned this tag  env:dev

- You can build a tag-search expression by using actual keys and key values, or by using the following wildcard characters.

`%` (percent sign): Matches any number of characters.

`_` (underscore): Matches one character.

Sample Expression	Description	Sample Search Result
'env': 'dev%'	Finds the tags with the key <code>env</code> and a value starting with <code>dev</code> , or the instances that are assigned such tags.	The following tags, or the instances that are assigned any of these tags:  env:dev env:dev1
'env': 'dev_'	Finds the tags with the key <code>env</code> and the value <code>devX</code> where <code>X</code> can be any one character, or finds the instances that are assigned such tags.	The following tags, or the instances that are assigned any of these tags:  env:dev1 env:dev2

- To use a single quotation mark (`'`), the percent sign (`%`), or the underscore (`_`) as a literal character in a search expression, escape the character by prefixing a backslash (`\`).

Sample Expression	Description	Sample Search Result
'env': 'dev\_%'	Finds the tags with the key <code>env</code> and a value starting with <code>dev_</code> , or the instances that are assigned such tags.	The following tags, or the instances that are assigned any of these tags:  <code>env:dev_1</code> <code>env:dev_admin</code>

- You can use the Boolean operators AND, OR, and NOT in your search expressions:

Sample Expression	Description	Sample Search Result
'env' OR 'owner'	Finds the tags with the key <code>env</code> or the key <code>owner</code> , or the instances that are assigned either of those keys.	The following tags, or the instances that are assigned <i>any of these tags</i> :  <code>env:dev</code> <code>owner:admin</code>
'env' AND 'owner'	Finds the instances that are assigned the tags <code>env</code> <i>and</i> <code>owner</code> .  <b>Note:</b> This expression won't return any results when used to search for tags, because a tag can have only one key.	The instances that are assigned <i>all of the following tags</i> :  <code>env:dev</code> <code>owner:admin</code>
NOT 'env'	Finds the tags that have a key other than <code>env</code> , or the instances that are assigned such tags.  <b>Note:</b> Untagged instances as well will satisfy this search expression.	The following tags, or the instances that are assigned <i>any of these tags</i> or no tags:  <code>owner:admin</code> <code>department</code>
('env' OR 'owner') AND NOT 'department'	Finds the tags that have the key <code>env</code> or the key <code>owner</code> but not the key <code>department</code> , or the instances that are assigned such tags.	The following tags, or the instances that are assigned <i>any of these tags</i> :  <code>env:dev</code> <code>owner:admin</code>



# 3

## Managing Network Access to MySQL Cloud Service

By default, network access to MySQL Cloud Service is provided by using SSH. The SSH connection uses the SSH key specified during the process to create a database deployment. By default, port 22 is used for SSH connections. To access other ports you must perform additional configuration tasks, such as creating an SSH tunnel or enabling access to the port.

When you create an MySQL Cloud Service database deployment in Oracle Cloud Infrastructure, network access to the deployment is provided and managed by the Oracle Cloud Infrastructure network components according to rules specified when you followed the instructions in [Prerequisites for Oracle Platform Services on Oracle Cloud Infrastructure](#) in the Oracle Cloud Infrastructure documentation.

When you create an MySQL Cloud Service database deployment in Oracle Cloud Infrastructure Classic, network access to the deployment is provided and managed using access rules in the MySQL Cloud Service console. With the exception of SSH access on port 22, the access rules created for a database deployment are disabled to ensure the deployment is secure by default. To open access to applications and management tools, you need to enable one of these access rules or create a custom access rule of your own.

### Topics

- [About Network Access to MySQL Cloud Service](#)
- [Generating a Secure Shell \(SSH\) Public/Private Key Pair](#)
- [Creating an SSH Tunnel to a Compute Node Port](#)
- [Enabling Access to a Compute Node Port](#)

## About Network Access to MySQL Cloud Service

By default, network access to the compute nodes associated with MySQL Cloud Service is provided by Secure Shell (SSH) connections on port 22. To access other network protocols and services requires additional configuration.

### SSH Access on Port 22

SSH is a cryptographic network protocol that uses two keys, one public and one private, to provide secure communication between two networked computers. Port 22 is the standard TCP/IP port that is assigned to SSH servers.

 **Note:**

Only the `opc` user can connect to MySQL Cloud Service over SSH. For more information on the `opc` user, see [Linux User Accounts](#).

When a database deployment is being created, the public key is specified on the Details page of the Create MySQL Cloud Service wizard by:

- Uploading the public key file.

When the public key file is uploaded during the database deployment creation process, the key is added to the Oracle Compute Cloud Service environment that supports MySQL Cloud Service and given a unique name.

- Entering the full hierarchical name of an existing public key using the format `/domain/user/keyname`.

An existing public key is one that has already been added to the Oracle Compute Cloud Service environment that supports MySQL Cloud Service, either as the result of an earlier database deployment creation process or by manual addition to Oracle Compute Cloud Service. To discover the full hierarchical name of a key, view the list of keys on the Oracle Compute Cloud Service SSH Public Keys page and hover over the short name of the key. For more information, see [Viewing an SSH Public Key in Using Oracle Compute Cloud Service](#).

For information on manually adding a public key to Oracle Compute Cloud Service, see [Adding an SSH Public Key in Using Oracle Compute Cloud Service](#).

When you access a compute node using SSH, you must provide the private key that matches the public key specified when the database deployment was created.

To generate the SSH public/private key pairs needed to access MySQL Cloud Service, see [Generating a Secure Shell \(SSH\) Public/Private Key Pair](#).

#### Access to Other Ports



This topic does not apply to Oracle Cloud Infrastructure.

To access network protocols and services on a compute node by using a port other than port 22, you must either:

- Enable network access to the port

You can use the MySQL Cloud Service console to enable access to a port on a compute node. For more information, see [Enabling Access to a Compute Node Port](#)

- Create an SSH tunnel to the port

Creating an SSH tunnel enables you to access a specific compute node port by using an SSH connection as the transport mechanism. To create the tunnel, you must have the SSH private key file that matches the public key specified during the database deployment creation process. For more information, see [Creating an SSH Tunnel to a Compute Node Port](#).

## Generating a Secure Shell (SSH) Public/Private Key Pair

Several tools exist to generate SSH public/private key pairs. The following sections show how to generate an SSH key pair on UNIX, UNIX-like and Windows platforms.

## Generating an SSH Key Pair on UNIX and UNIX-Like Platforms Using the ssh-keygen Utility

UNIX and UNIX-like platforms (including Solaris and Linux) include the ssh-keygen utility to generate SSH key pairs.

To generate an SSH key pair on UNIX and UNIX-like platforms using the ssh-keygen utility:

1. Navigate to your home directory:

```
$ cd $HOME
```

2. Run the ssh-keygen utility, where *filename* is your choice of file name for the private key:

```
$ ssh-keygen -b 2048 -t rsa -f filename
```

The ssh-keygen utility prompts you for a passphrase for the private key.

3. Enter a passphrase for the private key, or press Enter to create a private key without a passphrase:

Enter passphrase (empty for no passphrase): *passphrase*

### Note:

While a passphrase is not required, you should specify one as a security measure to protect the private key from unauthorized use. When you specify a passphrase, a user must enter the passphrase every time the private key is used.

The ssh-keygen utility prompts you to enter the passphrase again.

4. Enter the passphrase again, or press Enter again to continue creating a private key without a passphrase:

Enter the same passphrase again: *passphrase*

5. The ssh-keygen utility displays a message indicating that the private key has been saved as *filename* and the public key has been saved as *filename.pub*. It also displays information about the key fingerprint and randomart image.

## Generating an SSH Key Pair on Windows Using the PuTTYgen Program

The PuTTYgen program is part of PuTTY, an open source networking client for the Windows platform.

To generate an SSH key pair on Windows using the PuTTYgen program:

1. Download and install PuTTY or PuTTYgen.

To download PuTTY or PuTTYgen, go to <http://www.putty.org/> and click the **You can download PuTTY here** link.

2. Run the PuTTYgen program.  
The PuTTY Key Generator window is displayed.
3. Set the **Type of key to generate** option to **SSH-2 RSA**.
4. In the **Number of bits in a generated key** box, enter **2048**.
5. Click Generate to generate a public/private key pair.  
As the key is being generated, move the mouse around the blank area as directed.
6. (Optional) Enter a passphrase for the private key in the **Key passphrase** box and reenter it in the **Confirm passphrase** box.

 **Note:**

While a passphrase is not required, you should specify one as a security measure to protect the private key from unauthorized use. When you specify a passphrase, a user must enter the passphrase every time the private key is used.

7. Click **Save private key** to save the private key to a file. To adhere to file-naming conventions, you should give the private key file an extension of `.ppk` (PuTTY private key).
8. Select all of the characters in the **Public key for pasting into OpenSSH authorized\_keys file** box.  
Make sure you select all the characters, not just the ones you can see in the narrow window. If a scroll bar is next to the characters, you aren't seeing all the characters.
9. Right click somewhere in the selected text and select **Copy** from the menu.
10. Open a text editor and paste the characters, just as you copied them. Start at the first character in the text editor, and do not insert any line breaks.
11. Save the text file in the same folder where you saved the private key, using the `.pub` extension to indicate that the file contains a public key.
12. If you or others are going to use an SSH client that requires the OpenSSH format for private keys (such as the `ssh` utility on Linux), export the private key:
  - a. On the **Conversions** menu, choose **Export OpenSSH key**.
  - b. Save the private key in OpenSSH format in the same folder where you saved the private key in `.ppk` format, using an extension such as `.openssh` to indicate the file's content.

## Creating an SSH Tunnel to a Compute Node Port

To create an SSH tunnel to a port on a compute node associated with MySQL Cloud Service, you use Secure Shell (SSH) client software that supports tunneling.

Several SSH clients that support tunneling are freely available. The following sections show how to use SSH clients on the Linux and Windows platforms to connect to a compute node using an SSH tunnel.

## Creating an SSH Tunnel Using the ssh Utility on Linux

The Linux platform includes the ssh utility, an SSH client that supports SSH tunneling.

Before you use the ssh utility to create an SSH tunnel, you need the following:

- The IP address of the target compute node.  
The IP addresses associated with a database deployment on MySQL Cloud Service are listed on the details page associated with the database deployment. For instructions to display this page, see [Viewing Detailed Information for a MySQL Server Deployment](#).
- The SSH private key file that pairs with the public key used during the database deployment creation process.
- The port number for which you want to create an SSH tunnel.

To create an SSH tunnel for a port using the ssh utility on Linux:

1. In a command shell, set the file permissions of the private key file so that only you have access to it:

```
$ chmod 600 private-key-file
```

*private-key-file* is the path to the SSH private key file that matches the public key used during the database deployment creation process.

2. Run the ssh utility:

```
$ ssh -i private-key-file -L local-port:target-ip-address:target-port opc@target-ip-address
```

where:

- *private-key-file* is the path to the SSH private key file.
  - *local-port* is the number of an available port on your Linux system. Specify a port number greater than 1023 and less than 49152 to avoid conflicts with ports that are reserved for the system. As a good practice, and for the sake of simplicity, you should specify the same port number as the one to which you are creating a tunnel.
  - *target-ip-address* is the IP address of the target compute node in *x.x.x.x* format.
  - *target-port* is the port number to which you want to create a tunnel.
3. If this is the first time you are connecting to the target compute node, the ssh utility prompts you to confirm the public key. In response to the prompt, enter **yes**.

After the SSH tunnel is created, you can access the port on the target compute node by specifying *localhost:local-port* on your Linux system.

## Creating an SSH Tunnel Using the PuTTY Program on Windows

PuTTY is a freely available SSH client program for Windows that supports SSH tunneling.

Before you use the ssh utility to create an SSH tunnel, you need the following:

- The IP address of the target compute node.

The IP addresses associated with a database deployment on MySQL Cloud Service are listed on the details page associated with the database deployment. For instructions to display this page, see [Viewing Detailed Information for a MySQL Server Deployment](#).

- The SSH private key file that pairs with the public key used during the database deployment creation process.
- The port number for which you want to create an SSH tunnel.

To create an SSH tunnel for a port using the PuTTY program on Windows:

1. Download and install PuTTY.

To download PuTTY, go to <http://www.putty.org/> and click the **You can download PuTTY here** link.

2. Run the PuTTY program.

The PuTTY Configuration window is displayed, showing the Session panel.

3. Configure SSH connectivity:

- a. In **Host Name (or IP address)** box, enter the IP address of the target compute node.

- b. Confirm that the **Connection type** option is set to **SSH**.

- c. In the Category tree, expand **Connection** if necessary and then click **Data**.  
The Data panel is displayed.

- d. In **Auto-login username** box, enter **opc**.

- e. Confirm that the **When username is not specified** option is set to **Prompt**.

- f. In the Category tree, expand **SSH** and then click **Auth**.  
The Auth panel is displayed.

- g. Click the **Browse** button next to the **Private key file for authentication** box. Then, in the Select private key file window, navigate to and open the private key file that matches the public key used during the database deployment creation process.

4. Add a forwarded port:

- a. In the Category tree, click **Tunnels**.  
The Tunnels panel is displayed.

- b. In the **Source Port** box, enter the number of an available port on your system. Specify a port number greater than 1023 and less than 49152 to avoid conflicts with ports that are reserved for the system. As a good practice, and for the sake of simplicity, you should specify the same port number as the one to which you are creating a tunnel.

- c. In the **Destination box**, enter the IP address of the target compute node, a colon, and the port number to which you want to create a tunnel; for example, 192.0.2.100:1521.

- d. Confirm that the **Local** and **Auto** options are set.

- e. Click Add to add the forwarded port.

The new forwarded port appears in the **Forwarded ports** list.

5. In the Category tree, click **Session**.

The Session panel is displayed.

6. In the **Saved Sessions** box, enter a name for this connection configuration. Then, click **Save**.
7. Click **Open** to open the connection.

The PuTTY Configuration window is closed and the PuTTY window is displayed.

8. If this is the first time you are connecting to the target compute node, the PuTTY Security Alert window is displayed, prompting you to confirm the public key. Click **Yes** to continue connecting.

After the SSH tunnel is created, you can access the port on the target compute node by specifying `localhost:local-port` on your system, where `local-port` is the source port you specified when creating the tunnel.

## Enabling Access to a Compute Node Port



This topic does not apply to Oracle Cloud Infrastructure.

MySQL Cloud Service relies on Oracle Compute Cloud Service to provide secure network access to database deployments. You can use the MySQL Cloud Service console to perform network access operations such as enabling access to a port on a compute node, or creating new access rules.

When a database deployment is created, the following access rules are created:

- **ora\_p2admin\_ssh**, which controls public-internet access to port 22, the port used by SSH. Enabled by default.
- **ora\_p2admin\_mysql**, which controls public-internet access to port 3306, the port used by MySQL. Disabled by default.
- **ora\_trusted\_hosts\_mysql**, which defines a range of IP addresses with access to port 3306, the port used by MySQL. Enabled by default.
- **ora\_p2admin\_em**, which controls access to the default port for MySQL Enterprise Monitor. Disabled by default. To access MySQL Enterprise Monitor, this must be enabled.



### Note:

These access rules are system rules. As such, they can be enabled or disabled, but cannot be edited or deleted.

To enable access to a compute node port, you enable the appropriate security rule. When you enable one of the predefined security rules, the given port on the compute node is opened to the public internet. To enable access to a different port, or restrict access to a port, you must create a security rule. For general information about security rules, see *Using Oracle Compute Cloud Service*.


### Enabling Port Access by Enabling an Automatically Created Access Rule

You can use the MySQL Cloud Service console to enable one of the automatically created access rules:

1. Open the MySQL Cloud Service console.  
For detailed instructions, see [Accessing MySQL Cloud Service Console](#).
2. From the  menu for the database deployment, select **Access Rules**.  
The Access Rules page is displayed.
3. Locate the rule you want to enable.
4. From the  menu for the located rule, select **Enable**.  
The Enable Access Rule window is displayed.
5. Select **Enable**.  
The Enable Access Rule window closes and the rule is displayed as enabled in the list of rules. The given port on the compute node is opened to the public internet.

### Enabling or Restricting Port Access by Creating an Access Rule

You can create an access rule to enable ports not associated with a predefined rule, or to restrict access to ports to only permit connections from specific IP addresses:

1. Open the MySQL Cloud Service console.  
For detailed instructions, see [Accessing MySQL Cloud Service Console](#).
2. From the  menu for the database deployment, select **Access Rules**.  
The Access Rules page is displayed. For information about the details provided on this page, see [MySQL Cloud Service Access Rules Page](#).
3. Click **Create Rule**. In the Create Access Rule dialog, enter the following information.
  - **Rule Name:** Any name to identify this rule. Must start with a letter, followed by letters, numbers, hyphens, or underscores. Cannot start with `ora_` or `sys_`.
  - **Description:** Any description of your choice (optional).
  - **Source:** The hosts from which traffic should be allowed. Choices are:
    - **PUBLIC-INTERNET:** The `public-internet` IP List.
    - **custom:** A custom list of addresses from which traffic should be allowed. In the field that displays below when you select this option, enter a comma-separated list of the subnets (in CIDR format) or IPv4 addresses for which you want to permit access.
  - **Destination:** The security list to which traffic should be allowed. The only option is `mysql_MASTER`.
  - **Destination Port(s):** The port or range of ports you want to open. Specify a single port, such as 5001, or a range of ports separated by a hyphen, such as 5001-5010.
4. Click **Create**.  
The Create Access Rule dialog closes and the rule is displayed in the list of rules. New rules are enabled by default.



 **Tip:**

If necessary, adjust the number of results displayed on the Access Rules page so you can see the newly created rule.

# 4

## Administering MySQL Cloud Service

This section describes tasks for administering your MySQL Cloud Service environment and the MySQL databases contained therein.

### Topics

- [Adding Storage to MySQL Cloud Service](#)
- [Scaling MySQL Cloud Service](#)
- [Updating or Adding a New SSH Public Key](#)
- [Maintaining the Manageability of Your Database Deployments](#)
- [Loading Data into the MySQL Database](#)
- [Tuning Database Performance on MySQL Cloud Service](#)

## Adding Storage to MySQL Cloud Service

If a database deployment on MySQL Cloud Service is running out of storage, you can extend the existing storage volumes.

The following storage volumes can be extended:

- **Usable Database Storage**—storage available to the MySQL database. This storage volume can be extended up to 1024 GB.
- **Backup Storage**—storage available for backups stored on the compute node. This storage volume can be extended up to 2000 GB. This option is available only if backup was defined when the service was created.
- **Redolog Storage**—storage available for MySQL redo logging. This storage volume can be extended up to 1024 GB.

Extending a storage volume sets the storage volume to the defined value. It does not add the new value to the existing value, but sets the storage volume to the defined value. For example, if your current storage volume is 50GB and you define the new value as 100GB, the storage volume is extended to 100GB, not 150GB.

Each MySQL Cloud Service instance has 10 storage volumes assigned. Six of those are used, out-of-the-box, leaving 4 possible extensions.

### Note:

It is not possible to shrink a storage volume.

### Adding Storage

To extend a storage volume, do the following:

1. Open the **Instance Overview** page for the service by clicking the service name on the console.
2. Select the menu in the **Resources** section on the **Instance Overview** page.
3. Select **Add Storage** from the menu.  
The **Add Storage** dialog is displayed.
4. Define the required values and click **Yes, Add Storage** to extend your storage volumes to the defined value.  
The service restarts.

**Note:**

If you define a value which is larger than the defined limits, or lower than the current value, the field is highlighted in red and a tooltip is displayed explaining the error.

## Scaling MySQL Cloud Service



This topic does not apply to Oracle Cloud Infrastructure.

If a database deployment on MySQL Cloud Service is performing poorly or is running out of storage, you can scale up the environment supporting the database deployment.

Usually, the need to scale arises as the result of analyzing database performance, as described in [Tuning Database Performance on MySQL Cloud Service](#).

Occasionally, the need to scale arises from some change made to the database or backup configuration after it was created. For example, you might need to scale up the compute shape to one of the high-memory options, such as scaling from "OC4 - 2 OCPU, 15 GB RAM" to "OC2M - 2 OCPU, 30 GB RAM".

### Scaling the Compute Shape for MySQL Cloud Service

**Note:**

When you scale the compute shape of a database deployment on MySQL Cloud Service, the deployment is put into Maintenance status during the operation and it is restarted. As a result of the restarting, any resources you've manually added using the Compute Cloud Service console become detached from the database deployment. For more information and for instructions on reattaching such resources, see [About Stopping, Starting and Restarting a MySQL Server Deployment](#).

To scale the compute shape for a database deployment:

1. Open the **Instance Overview** page for the service by clicking the service name on the console.
2. Select the menu in the **Resources** section on the **Instance Overview** page.


3. Select **Scale Up/Down** from the menu.  
The **Scale Up/Down** dialog is displayed.
4. Choose the required shape and click **Yes, Scale Up/Down VM** to scale your instance.  
The service restarts.

## Updating or Adding a New SSH Public Key

Should the need arise, you can add SSH public keys for the `opc` and `oracle` users to the compute nodes associated with a database deployment on MySQL Cloud Service.

After you add the public key, you can provide the matching private key to connect as the `opc` user.

To add an SSH public key:

1. Open the MySQL Cloud Service console.  
For detailed instructions, see [Accessing MySQL Cloud Service Console](#).
2. From the  menu for the database deployment, select **SSH Access**.  
The **Add New Key** overlay is displayed with its **Key value** field displaying the most recent SSH public key.
3. Specify the new public key using one of the following methods:
  - Select **Upload a new SSH Public Key value** and click **Choose File** to select a file that contains the public key.
  - Select **Key value**. Delete the current key value and paste the new public key into the text area. Make sure the value does not contain line breaks or end with a line break.
4. Click **Add New Key**.

You can also add SSH public keys to one or more deployments on the SSH Access page. For information, see [MySQL Cloud Service SSH Access Page](#).

## Maintaining the Manageability of Your Database Deployments

The following best practices will ensure that your database deployments on MySQL Cloud Service stay manageable.

To keep your database deployments manageable, follow these guidelines:

- Do not disable or close access to the SSH port (port 22).  
You can open other ports and protocols. See [About Network Access to MySQL Cloud Service](#) for information about the available options.
- Do not detach, change file access permissions for, or change the mount point of any storage volume attached to a compute node during the creation of your database deployment. In particular, do not unmount or change the file access permissions of `/u01` through `/u05`.

- Do not change compute node OS users and SSH key settings that were configured during the creation of your database deployment.
- Apply **only** patches that are available through MySQL Cloud Service. Do **not** apply patches from any other source unless directed to by Oracle Support.

## Loading Data into the MySQL Database

You load data into a MySQL database on MySQL Cloud Service using the same tools you would use for a MySQL database on your system or in your data center. The fact that the database is housed in the Oracle Cloud does not place any restrictions on data loading.

However, transmission speeds across the Internet tend to be slower, sometimes much slower, than on internal networks, and any data loading approach that you use should factor this in. For the sake of efficiency, therefore, it is recommended you move the files containing the data on to a compute node associated with MySQL Cloud Service using a single bulk transfer before the data-loading operation is performed. Consequently, the processing associated with the data loading operation occurs within the MySQL Cloud Service environment and proceeds more rapidly.

For more information on how to load data into your MySQL Server, see [MySQL 5.7 Reference Manual](#).



### Note:

It is also possible to instantiate a MySQL Cloud Service deployment using a backup stored in a cloud container. For more information, see [Creating a MySQL Server Deployment](#).

## Tuning Database Performance on MySQL Cloud Service

You tune the performance of MySQL Database on MySQL Cloud Service using the same tools you would use for MySQL database running on any system in your data center. The fact that the database is housed in the Oracle Cloud does not place any restrictions on performance tuning.

# 5

## Accessing MySQL Cloud Service

This section describes how to access tools, utilities and interfaces available in MySQL Cloud Service.

### Note:

When you create a MySQL Cloud Service database deployment, you provide a Secure Shell (SSH) public key value so that you can directly access command-line tools on a compute node of the deployment. For information about creating an SSH public key value, see [Generating a Secure Shell \(SSH\) Public/Private Key Pair](#).

### Topics

- [Connecting to a Compute Node Through Secure Shell \(SSH\)](#)
- [Generating a Secure Shell \(SSH\) Public/Private Key Pair](#)

## Connecting to a Compute Node Through Secure Shell (SSH)

To gain local access the tools, utilities and other resources on a compute node associated with MySQL Cloud Service, you use Secure Shell (SSH) client software to establish a secure connection and log in as the user `opc`. Once logged in as `opc`, you can switch to the `oracle` user.

Several SSH clients are freely available. The following sections show how to use SSH clients on UNIX, UNIX-like and Windows platforms to connect to a compute node associated with MySQL Cloud Service.

## Connecting to a Compute Node Using the ssh Utility on UNIX and UNIX-Like Platforms

UNIX and UNIX-like platforms (including Solaris and Linux) include the `ssh` utility, an SSH client.

### Before You Begin

Before you use the `ssh` utility to connect to a compute node, you need the following:

- The IP address of the compute node

The IP address of a compute node associated with a database deployment on MySQL Cloud Service is listed on the MySQL Cloud Service Overview page. For

instructions to display this page, see [Viewing Detailed Information for a MySQL Server Deployment](#).

- The SSH private key file that matches the public key associated with the deployment.

### Procedure

To connect to a compute node using the `ssh` utility on UNIX and UNIX-like platforms:

1. In a command shell, set the file permissions of the private key file so that only you have access to it:

```
$ chmod 600 private-key-file
```

*private-key-file* is the path to the SSH private key file that matches the public key that is associated with the deployment.

2. Run the `ssh` utility:

```
$ ssh -i private-key-file user-name@node-ip-address
```

where:

- *private-key-file* is the path to the SSH private key file.
  - *user-name* is the **opc** user
    - Connect as the user `opc` to perform operations that require root access to the compute node, this user can use the `sudo su oracle` command to use MySQL from the command prompt..
  - *node-ip-address* is the IP address of the compute node in `x.x.x.x` format.
3. If this is the first time you are connecting to the compute node, the `ssh` utility prompts you to confirm the public key. In response to the prompt, enter **yes**.

## Connecting to a Compute Node Using the PuTTY Program on Windows

PuTTY is a freely available SSH client program for Windows.

### Before You Begin

Before you use the PuTTY program to connect to a compute node, you need the following:

- The IP address of the compute node

The IP address of a compute node associated with a database deployment on MySQL Cloud Service is listed on the MySQL Cloud Service Overview page. For instructions to display this page, see [Viewing Detailed Information for a MySQL Server Deployment](#).
- The SSH private key file that matches the public key associated with the deployment. This private key file must be of the PuTTY `.ppk` format. If the private key file was originally created on the Linux platform, you can use the PuTTYgen program to convert it to the `.ppk` format.

### Procedure

To connect to a compute node using the PuTTY program on Windows:

1. Download and install PuTTY.  
To download PuTTY, go to <http://www.putty.org/> and click the **You can download PuTTY here** link.
2. Run the PuTTY program.  
The PuTTY Configuration window is displayed, showing the Session panel.
3. In **Host Name (or IP address)** box, enter the IP address of the compute node.
4. Confirm that the **Connection type** option is set to **SSH**.
5. In the Category tree, expand **Connection** if necessary and then click **Data**.  
The Data panel is displayed.
6. In **Auto-login username** box, enter the **opc** user to connect as.
7. Confirm that the **When username is not specified** option is set to **Prompt**.
8. In the Category tree, expand **SSH** and then click **Auth**.  
The Auth panel is displayed.
9. Click the **Browse** button next to the **Private key file for authentication** box.  
Then, in the Select private key file window, navigate to and open the private key file that matches the public key that is associated with the deployment .
10. In the Category tree, click **Session**.  
The Session panel is displayed.
11. In the **Saved Sessions** box, enter a name for this connection configuration. Then, click **Save**.
12. Click **Open** to open the connection.  
The PuTTY Configuration window is closed and the PuTTY window is displayed.
13. If this is the first time you are connecting to the VM, the PuTTY Security Alert window is displayed, prompting you to confirm the public key. Click **Yes** to continue connecting.



# 6

## Backing Up and Restoring Databases on MySQL Cloud Service

This section explains how to back up and restore MySQL databases on MySQL Cloud Service.

### Note:

MySQL Cloud Service uses MySQL Enterprise Backup for backup operations. For more information, see [MySQL Enterprise Backup User Manual](#).

### Topics

- [About Backing Up Databases on MySQL Cloud Service](#)
- [Creating an On-Demand Backup](#)
- [Deleting a Backup](#)
- [Customizing the Current Backup Configuration](#)
- [Restoring from a Specific Backup](#)
- [Restoring to a Specific Point in Time](#)
- [Updating the Password for Backing Up to the Storage Cloud](#)

## About Backing Up Databases on MySQL Cloud Service

By backing up your MySQL Cloud Service database, you can protect the database against loss if a failure occurs. By restoring from a backup, you can restore that database to its state at the time of the backup.

MySQL Cloud Service provides a backup feature that backs up:

- The complete contents of the MySQL data directory. For more information on the files backed up, see [Files Backed Up](#).

To provide this backup feature, MySQL Cloud Service relies on MySQL Enterprise Backup which is installed in the database deployment.

### Note:

MySQL backups are not encrypted.

## Default Backup Configuration

When you create a database deployment, you choose one of the following backup destinations:

- **Both Cloud Storage and Local Storage.** Backups are configured to be created automatically and stored both on local compute node storage and on an Oracle Storage Cloud Service container. You can use an existing container or create one when creating the deployment.
- **Cloud Storage Only.** Backups are configured to be created automatically and stored on an Oracle Storage Cloud Service container. You can use an existing container or create one when creating the deployment.
- **None.** No backup configuration is created.

The backup configuration created when you choose a destination other than **None** follows a set of Oracle best-practice guidelines:

- Full backup of the contents of the database server's data directory, followed by rolling incremental backups on a seven-day cycle.
- Automatic backups daily, the specific time is configured in the **Configure Backups** dialog of the service administration page.
- Retention period:
  - **Both Cloud Storage and Local Storage:** 30 days, with the 7 most recent days' backups available on local storage. This set of backups consists of one Full backup and 6 incremental backups, based on the Full backup. After 7 days, a new Full backup is created, and the preceding Full Backup, and its associated incremental backups, are deleted.
  - **Cloud Storage Only:** 30 days..

If the defaults do not suit your needs, you can customize the backup configuration for your database deployment. For information, see [Customizing the Current Backup Configuration](#).

## About Local Storage for Backups

When a database deployment is created on MySQL Cloud Service, Oracle Compute Cloud Service storage volumes are created and associated with the compute nodes.

You can see details about storage volumes in the Compute Cloud Service console. See [Viewing a Storage Volume in Using Oracle Compute Cloud Service](#).

The space allocated for the local storage used for backups is twice the space allocated for data storage. For example, if you chose 100 GB for Usable Database Storage when creating the database deployment, 200 GB are allocated for backups.

## When Do Backups Occur

If backups are configured for a service instance, they are **scheduled** to occur automatically. You can also initiate an **on-demand** backup immediately without having to wait for the next scheduled backup.

By default, scheduled backups occur at the following times:

- Full backups are initiated weekly starting 12 hours after backups were enabled on a service instance, rounded to the nearest five-minute interval.

For example, if a service instance is created with backups at 1:01 PM on a Monday, full backups are initiated at 1:00 AM on Tuesdays.

- Incremental backups are initiated every day, except the day of a full backup, at the same time that full backups are initiated.

For example, if a service instance is created with backups at 1:01 PM on a Monday, incremental backups are initiated at 1:00 AM every day except Tuesdays.

- If you stop a service instance, and then restart it after the next scheduled backup, the overdue backup will be immediately.

You can change the default schedule on which automated backups are initiated.

After configuring backups for a service instance, you can also enable or disable the backup service as needed.

MySQL Cloud Service automatically disables backups on a service instance if the following conditions are true:

- You specified a notification email address when you created the service instance.
- Three consecutive failures of the same type occur with scheduled backups.

Both you and the account administrator will receive an email notification each time a backup fails and also when backups are automatically disabled. A reminder notification is sent every week until backups are enabled again.

### Viewing Available Backups in the Console

To view the available backups in the console, do the following:

1. Open the console.
2. Select the service. The service overview page is displayed.
3. Select **Administration**. The service administration page is displayed.
4. Select the Backup tab to view the list of available backups.

## Creating an On-Demand Backup

You can create an on-demand backup of a MySQL Cloud Service database deployment by using the MySQL Cloud Service console. See [Creating an On-Demand Backup by Using the MySQL Cloud Service Console](#).

## Creating an On-Demand Backup by Using the MySQL Cloud Service Console

You can create an on-demand backup of a MySQL Cloud Service database deployment from the MySQL Cloud Service Backup page.

1. Open the MySQL Cloud Service console.  
For detailed instructions, see [Accessing MySQL Cloud Service Console](#).
2. Click the database deployment for which you want to create a backup.  
The MySQL Cloud Service Overview page is displayed.
3. Click the Administration tile.

The MySQL Cloud Service Backup page is displayed.

4. Click **Backup Now** and then confirm the action.

The backup process begins.

## Deleting a Backup

### Deleting a Backup from the MySQL Cloud Service Console



#### Note:

It is not possible to delete an automatic backup. Automatic backups are deleted during a scheduled backup after their expiration date.

To delete an on-demand backup, do the following:

1. Navigate to the required service in the console.
2. Select Administration to open the Backup tab.
3. Click the menu adjacent to the backup you want to delete.
4. Click Delete.

The backup is deleted.

## Customizing the Current Backup Configuration

This section describes how to customize your backup configuration from the MySQL Cloud Service Console. The following tasks are possible:

- Defining the schedule for a full backup. The full backup is performed once a week.
- Defining the schedule for an incremental backup. Incremental backups are performed daily.
- Defining the default retention period. This defines for how many days the backups are kept in the cloud storage. Backups older than the defined number of days are deleted.

### Customizing Backup Configuration From the Console

To customize the backup configuration, do the following:

1. Select the service.
2. On the service details page, select **Administration** to open the **Backup** tab.
3. Select **Configure Backups** to open the **Configure Backups** dialog.
4. Define the schedules for your backups and the retention period.

## Restoring from a Specific Backup


You can restore a specific backup and perform recovery to that backup on a MySQL Cloud Service database deployment by using the MySQL Cloud Service console. See [Restoring from a Specific Backup by Using the MySQL Cloud Service Console](#).

 **Note:**

If MySQL Transparent Data Encryption (TDE) is enabled on the MySQL instance, it is not currently possible to restore a backup of that instance.

## Restoring from a Specific Backup by Using the MySQL Cloud Service Console

You can use the MySQL Cloud Service console to restore a specific backup and perform recovery to that backup:

1. Open the MySQL Cloud Service console.  
For detailed instructions, see [Accessing MySQL Cloud Service Console](#).
2. Click the database deployment you want to restore and recover.  
The MySQL Cloud Service Overview page is displayed.
3. Click the Administration tile.  
The MySQL Cloud Service Backup page is displayed.
4. In the list of backups, locate the backup you want to restore from.
5. In the entry for the backup you want to restore from, click the  menu, choose **Restore** and then confirm the action.

The restore process performs these steps:

- Shuts down the database
- Prepares for restore
- Performs the restore
- Restarts the database instance after restore

## Restoring to a Specific Point in Time

You can restore from a backup and perform recovery to a specific point in time on a database deployment by using the MySQL Cloud Service console. See [Restoring to a Specific Point in Time by Using the MySQL Cloud Service Console](#).



**Note:**

If MySQL Transparent Data Encryption (TDE) is enabled on the MySQL instance, it is not currently possible to restore a backup of that instance.

It is also possible to view and refine the Restore History, using the filters on the **Restore History** section of the **Backup** tab.

## Restoring to a Specific Point in Time by Using the MySQL Cloud Service Console

You can use the MySQL Cloud Service console to restore from a backup and perform recovery to a specific point in time:

1. Open the MySQL Cloud Service console.  
For detailed instructions, see [Accessing MySQL Cloud Service Console](#).
2. Click the database deployment you want to restore and recover.  
The MySQL Cloud Service Overview page is displayed.
3. Click the Administration tile.  
The MySQL Cloud Service Backup page is displayed.
4. Click **Restore Point in Time**.  
The **Restore Point in Time** dialog is displayed.
5. In the list of recovery options, select **Point in Time** to indicate how you want to specify the end point of the recovery operation. Then, enter the appropriate value.
6. Click **Restore Point in Time**.  
The restore process performs these steps:
  - Shuts down the database
  - Prepares for restore
  - Performs the restore
  - Restarts the database instance after restore

## Updating the Password for Backing Up to the Storage Cloud

Because Oracle Cloud requires users to change their passwords on a regular basis, you need to perform this task regularly.

You can update the password by using the MySQL Cloud Service console.

### Updating the Password by Using the MySQL Cloud Service Console

1. Open the MySQL Cloud Service console.
2. Click the name of the MySQL Cloud Service deployment whose backup credentials you want to update.  
The **Overview** page for the deployment is displayed.

3. Click the **Manage this service** menu adjacent the service name, and select **Service Credentials**.

The **Update Service Credentials** dialog is displayed.

4. Enter the Cloud user name and new password.
5. Click **Update** and confirm the operation.

# 7

## Patching MySQL Cloud Service

This section explains how to apply a patch to MySQL Cloud Service, and roll back the patch as necessary.

### Topics

- [About Patching an Oracle MySQL Cloud Service Instance](#)
- [Checking Prerequisites Before Applying a Patch](#)
- [Applying a Patch](#)
- [Rolling Back a Patch](#)

## About Patching an Oracle MySQL Cloud Service Instance

You can quickly and easily apply patches to a MySQL Cloud Service instance from the user interface.

Once you navigate to the Patching page on the Oracle MySQL Cloud Service user interface, applying a patch to service instance is a one-click operation. See [Patching MySQL Cloud Service](#).

### What kinds of patches are available

You can patch the MySQL Server of an Oracle MySQL Cloud Service instance.

This patch is applied from the Patching page. See [Patching MySQL Cloud Service](#).

### About Operating System (OS) Patching

MySQL Cloud Service does not provide cloud tooling for OS patching. You are responsible for installing OS patches to existing service instances.

You can obtain Oracle Linux OS patches from the Oracle's Unbreakable Linux Network if you have an Oracle Linux support subscription. You can also obtain Linux OS patches from the Oracle Linux Public yum server: <http://public-yum.oracle.com>.

### When to Apply Patches

New approved patches are available and displayed on the Patching page roughly on a quarterly basis. Apply the most recent patches promptly. Delaying patches could cause your service to be unsupported for future patching and upgrade.

### What Happens When Patching Starts

As patching starts, the patching operation first performs internal prechecks. If the prechecks fail, the patching operation will fail and leave the service instance untouched.

The prechecks check for the following:

- Disk space shortage



- Missing database connectivity
- Servers not running
- Storage access failure

The prechecks do not check whether another administration task (backup, restoration, or scaling) is in progress, which would prevent patching.

You can perform prechecks without attempting to patch, and first remedy any problems found. For more information, see [Checking Patch Prerequisites](#).

An automatic backup to use for rollback and restoration is created only after patching prechecks succeed. If you need to restore the state of the service instance, use the backup and run patching again.



**Note:**

If automatic backup fails, then the patching operation fails and does not apply the patch.

### What Happens to the Service During Patching

The MySQL Server is stopped for the duration of the patching operation. The MySQL Server is restarted when the patching operation is finished.

## Checking Prerequisites Before Applying a Patch

Before you apply a patch, you can check its prerequisites to make sure that it can be successfully applied.

The prerequisites-checking operation:


- Confirms that the patch is available for download.
- Verifies that there is enough space to apply the patch.
- Compares the patch's prerequisites to the database deployment.

You can perform the prerequisites-checking operation using the MySQL Cloud Service console. See [Checking Patch Prerequisites](#).

## Checking Patch Prerequisites

You can use the MySQL Cloud Service Patching page to check the prerequisites of a patch before you apply it to a database deployment .

1. Open the MySQL Cloud Service console.  
For detailed instructions, see [Accessing MySQL Cloud Service Console](#).
2. Click the deployment to which you want to apply a patch.  
The MySQL Cloud Service Overview page is displayed.
3. Click the Administration tile, and then click the Patching tab.  
The MySQL Cloud Service Patching page is displayed. A list of patches you can apply appears in the Available Patches section.

4. In the entry for the patch whose prerequisites you want to check, click the  menu and then select **Precheck**.

If you have previously checked prerequisites on the selected patch, the Patch Precheck Service window displays, showing the results of the previous check and asking you to confirm another precheck. In this case, click **Precheck** to continue.

The Patching page redisplay, showing a status message that your request has been submitted.

5. Refresh the Patching page occasionally to update the status message.

Note that the checking operation can take several minutes to complete.

When the checking operation completes, the status message displays the results of the prerequisite checking.

## Applying a Patch

You can apply a patch to a database deployment by using the MySQL Cloud Service console.

### Applying a Patch Using the MySQL Cloud Service Console

You can apply a patch to a database deployment on MySQL Cloud Service by using the MySQL Cloud Service Patching page.

#### Procedure

To apply a patch to a database deployment:

1. Open the MySQL Cloud Service console.


For detailed instructions, see [Accessing MySQL Cloud Service Console](#).

2. Click the deployment to which you want to apply a patch.

The MySQL Cloud Service Overview page is displayed.

3. Click the Administration tile and then click the Patching tab.

The MySQL Cloud Service Patching page is displayed. A list of patches you can apply appears in the Available Patches section.

4. In the entry for the patch you want to apply, click the  menu and then select **Patch**.

The Patch Service window closes and the patching operation begins. The patch process first performs a full backup before patching the database.

The Administration tile shows the starting time of the patching operation and a **Patching...** message replaces the **Patch** button.

When the patching operation completes, the Patching page shows the completion time of the patching operation, and a log of the operation's activities appears in the Details of Last Patching Activity section. If the operation was successful, the patch is removed from the list of patches in the Available Patches. If the operation failed, the patch remains in the list. In this case, check the Details of Last Patching Activity section for information about the failure.

 **Note:**

Patching operations are performed with a minimum of impact on the functioning of the database. However, during part of the operation the database restarts, making it inaccessible for a period of time .

## Rolling Back a Patch

You can roll back a patch on a database deployment by using the MySQL Cloud Service console. See [Rolling Back a Patch by Using the MySQL Cloud Service Console](#).

### Rolling Back a Patch by Using the MySQL Cloud Service Console

You can roll back the last patch to a database deployment on MySQL Cloud Service from the MySQL Cloud Service Patching page.

To roll back the last patch by using the MySQL Cloud Service console:

1. Open the MySQL Cloud Service console.  
For detailed instructions, see [Accessing MySQL Cloud Service Console](#).
2. Click the database deployment on which you want to roll back a patch.  
The MySQL Cloud Service Overview page is displayed.
3. Click the Administration tile and then click the Patching tab.  
The MySQL Cloud Service Patching page is displayed.
4. Click **Rollback**.

The Patching page redisplay, showing a status message that your request has been submitted, the Administration tile shows the starting time of the rollback operation, and a **Rolling back...** message replaces the **Rollback** button.

 **Note:**

Rollback operations are performed with a minimum of impact on the functioning of the database. However, during part of the operation the database is shut down, making it inaccessible. The database restarts when that portion of the rollback operation completes.

5. Refresh the Patching page occasionally to update the status message.

Note that the rollback operation can take several minutes to complete.

When the rollback operation completes, the Administration tile shows the completion time of the operation, and a log of the operation's activities appears in the Details of Last Patching Activity section.

# 8

## Migrating MySQL Databases to MySQL Cloud Service

You can migrate your on-premises MySQL databases to MySQL Cloud Service using various different approaches based on different tools and technologies.

### Topics

- [Migrating MySQL Data](#)

## Migrating MySQL Data

You can migrate your on-premises MySQL Database database to a MySQL Cloud Service using the method described here.

Some of the characteristics and factors to consider when choosing a migration method are:

- On-premises database version
- Oracle MySQL Cloud Service database version
- On-premises host operating system and version
- Quantity of data, including indexes
- Storage for data staging
- Acceptable length of system outage
- Network bandwidth

To migrate your data from an on-premises MySQL database to a MySQL Cloud Service database, it is strongly recommended to export your local database and copy the dump to the compute node before importing into the MySQL Cloud Service.

1. Back up your local server, preferably using MySQL Enterprise Backup. For more information, see [Backing Up a Database Server](#).
2. Copy the backup to your cloud service, using rsync, or scp for example.
3. Import the backup to your cloud service's MySQL Server, using the cloud service installation of MySQL Enterprise Backup. For more information on importing a backup, see [Restoring a Database](#).

# A

## Characteristics of a Newly Created Deployment

This section provides information about the content and configuration of a newly created database deployment on MySQL Cloud Service.

### Topics

- [Linux User Accounts](#)
- [Locations of Installed Software](#)

## Linux User Accounts



This topic does not apply to Oracle Cloud Infrastructure.

This section provides information about Linux user accounts that are provisioned on a MySQL Cloud Service deployment.

The MySQL Cloud Service compute node is provisioned with the following operating system user accounts.

User	Description
<code>opc</code>	The system administrator account you use with the <code>sudo</code> command to perform operations that require root-user access.
<code>oracle</code>	The database administrator account you use to access the system and perform non-root, database administration tasks. A home directory, <code>/u01/app/oracle/tools/home/oracle</code> , is created for this user. This user cannot use the <code>sudo</code> command to perform operations that require root-user access.
<code>root</code>	The root administrator for the system. You do not have direct access to this account. To perform operations that require root-user access, use the <code>sudo</code> command as the <code>opc</code> user.

### Note:

MySQL Server, and all associated processes, run under the `oracle` user. To run the `mysql` command line, `mysqladmin`, and so on, you must be logged in as the `oracle` user.

The following environment variables are created for the `opc` and `oracle` users.

Variable	Description
HOME	The home directory of the user, either <code>/home/opc</code> or <code>/u01/app/oracle/tools/home/oracle</code> .
HOSTNAME	The host name of the compute node.
MYSQL_HOME	The MySQL Server home directory: <code>/u01/bin/mysql</code> . Set for the <code>oracle</code> user, only.
PATH	The paths to search for executables; set to include: <ul style="list-style-type: none"> <li><code>/sbin</code></li> <li><code>/usr/sbin</code></li> <li><code>/bin</code></li> <li><code>/usr/bin</code></li> <li><code>\$MYSQL_HOME/bin</code></li> <li><code>\$HOME/bin</code></li> </ul> <p><code>mysqladmin</code> and <code>mysql.server</code> are available to the <code>oracle</code> user. To invoke the <code>mysql</code> command line, enter <code>mysql</code> on the command line as the <code>oracle</code> user. The <code>oracle</code> user is automatically logged in to the server.</p>
SHELL	The default shell, <code>/bin/bash</code> .
USER	The user name, either <code>opc</code> or <code>oracle</code> .

## Locations of Installed Software

This section provides information about the locations of installed software on a newly created MySQL Cloud Service database deployment.

When a database deployment is created, software is installed in the following locations.

Software	Installation Location
MySQL Database Server	<code>\$MYSQL_HOME: /u01/bin/mysql</code>

# B

## Oracle Cloud Pages for Administering MySQL Cloud Service

This section provides information about what you can do and what you see on each of the pages for administering MySQL Cloud Service.

### Topics

- [MySQL Cloud Service Instances Page](#)
- [MySQL Cloud Service Activity Page](#)
- [MySQL Cloud Service SSH Access Page](#)
- [MySQL Cloud Service IP Reservations Page](#)
- [MySQL Cloud Service Overview Page](#)
- [MySQL Cloud Service Access Rules Page](#)
- [MySQL Cloud Service Backup Page](#)
- [MySQL Cloud Service Patching Page](#)
- [MySQL Cloud Service Snapshots Page](#)
- [Create MySQL Cloud Service: Service Page](#)
- [Create MySQL Cloud Service: Service Details Page](#)
- [Create MySQL Cloud Service: Confirmation Page](#)

## MySQL Cloud Service Instances Page

The MySQL Cloud Service Instances page displays all MySQL Cloud Service deployments.




### What You Can Do From the MySQL Cloud Service Services Page

Use the MySQL Cloud Service Instances page to perform the tasks described in the following topics:

- [Viewing All MySQL Server Deployments](#)
- [Creating a MySQL Server Deployment](#)
- [Viewing Detailed Information for a MySQL Server Deployment](#)
- [Deleting a MySQL Server Deployment](#)

### What You See on the MySQL Cloud Service Instances Page

The following table describes the key information shown on the MySQL Cloud Service Instances page.

Element	Description
	Navigation menu providing access to the <b>Dashboard</b> and any other Oracle Cloud services in the identity domain.
<b>Instances</b>	Opens the <b>Instances</b> page, which enables you to create, edit, or delete instances.
<b>Activity</b>	Opens the <b>Activity</b> page which enables you to view all activity in your identity domain.
<b>SSH Access</b>	Opens the <b>SSH Access</b> page which enables you to view and update the SSH keys used by your instances.
<b>Summary</b>	Displays an overview of the running instances. <ul style="list-style-type: none"> <li>• <b>Instances</b>—Total number of configured deployments.</li> <li>• <b>OCPUs</b>—Total number of Oracle CPUs allocated across all deployments.</li> <li>• <b>Memory</b>—Total amount of compute node memory (RAM) allocated across all deployments.</li> <li>• <b>Storage</b>—Total amount of storage allocated across all deployments.</li> <li>• <b>Public IPs</b>—Number of public IP addresses allocated across all deployments.</li> </ul>
 (on right side of instance entry)	<ul style="list-style-type: none"> <li>• <b>Enterprise Monitor URL</b>—opens the MySQL Enterprise Monitor page in a new tab. This item is displayed if the MySQL Enterprise Monitor is configured.</li> <li>• <b>Start</b>—starts the instance, if stopped.</li> <li>• <b>Stop</b>—stops the instance, if running.</li> <li>• <b>Restart</b>—restarts the instance, if stopped or running.</li> <li>• <b>Access Rules</b>—Opens the Access Rules page.</li> <li>• <b>SSH Access</b>—Add an SSH public key to the deployment. See <a href="#">Updating or Adding a New SSH Public Key</a>.</li> <li>• <b>Delete</b>—Deletes the instance.</li> </ul>
<input type="text" value="Search by service name"/> 	Enter a full or partial deployment name to filter the list of deployments to include only those that contain the string in their name.
<b>Create Instance</b>	Click to create a new database deployment on MySQL Cloud Service. See <a href="#">Creating a MySQL Server Deployment</a> .
<b>Status</b>	Status of the deployment if it is not running. Statuses include “In Progress”, “Maintenance”, “Stopped”, and “Terminating”.
<b>Version</b>	Version of MySQL Server configured on the deployment. For example: 5.7.20
<b>Created On or Submitted On</b>	Date the deployment was created. During the creation process, the date when the creation request was submitted.
<b>OCPUs</b>	Number of Oracle CPUs allocated to the deployment.
<b>Memory</b>	Amount of compute node memory in GBs allocated to the deployment.
<b>Storage</b>	Amount of storage in GBs allocated to the deployment.
<b>Instance create and delete history</b>	Listing of attempts to create or delete a deployment. Click the triangle icon next to the title to view the history listing.



# MySQL Cloud Service Activity Page

The Activity page displays activities for all deployments in your identity domain. You can restrict the list of activities displayed using search filters.

## Topics

- [What You Can Do From the Activity Page](#)
- [What You See on the Activity Page](#)

## What You Can Do From the Activity Page

Use the Activity page to view operations for all MySQL Cloud Service deployments in your identity domain.

You can use the page's Search Activity Log section to filter the list of displayed operations based on:

- The time the operation was started
- The status of the operation
- The name of the deployment on which the operation was performed
- The type of the operation


In the table of results, you can:

- Click any column heading to sort the table by that column.
- Click the triangle at the start of an operation's row to see more details about that operation.

## What You See on the Activity Page

The following table describes the key information shown on the MySQL Cloud Service Activity page.

Element	Description
Start Time Range	Filters activity results to include only operations started within a specified time range. The range defaults to the previous 24 hours.
Status	Filters operations by status of the operation: <ul style="list-style-type: none"> <li>• All</li> <li>• Scheduled</li> <li>• Running</li> <li>• Succeeded</li> <li>• Failed</li> </ul> You can select any subset of status types. The default value is All.
Service Name	Filters the activity results to include operations only for the specified service instance. You can enter a full or partial service instance name.
Service Type	Filters the activity results to include operations only for instances of the specified service type. The default value is the current cloud service.
Operation	Filters the activity results to include selected types of operations. You can select any subset of the given operations. The default value is All.

Element	Description
<b>Search</b>	Searches for activities by applying the filters specified by the Start Time Range, Status, Service Name, Service Type and Operation fields, and displays activity results in the table.
<b>Reset</b>	Clears the Start Time Range and Service Name fields, and returns the Status and Operation fields to their default values.
Results per page	Specifies the number of results you want to view per page. The default value is 10.
	Displays status messages for the given operation. Clicking on the resulting downward arrow hides the status messages.
Service Name	Shows the name of the service instance and its identity domain: <i>service_instance:identity_domain</i> You can sort the column in ascending or descending order.
Service Type	Shows the type of cloud service for this instance. You can sort the column in ascending or descending order.
Operation	Shows the type of operation performed on the service instance. You can sort the column in ascending or descending order.
Status	Shows the status of the operation performed on the service instance. You can sort the column in ascending or descending order.
Start Time	Shows the time the operation started. You can sort the column in ascending or descending order.
End Time	Shows the time the operation ended, if the operation is complete. You can sort the column in ascending or descending order.
Initiated By	Shows the user that initiated the operation. The user can be any user in the identity domain who initiated the operation or, for certain operations such as automated backup, System. You can sort the column in ascending or descending order.

## MySQL Cloud Service SSH Access Page

The SSH Access page enables you to view and add SSH public keys to MySQL Cloud Service deployments in your identity domain. You can restrict the list of deployments displayed using search filters.

### Topics

- [What You Can Do From the Activity Page](#)
- [What You See on the Activity Page](#)

### What You Can Do From the SSH Access Page

Use the SSH Access page to view and add SSH public keys to MySQL Cloud Service deployments in your identity domain.


You can use the page's Search section to filter the list of displayed deployments based on deployment name.

In the table of results, you can:

- Click any column heading to sort the table by that column.
- Click the triangle at the start of a deployment's row to see more details.

### What You See on the SSH Access Page

The following table describes the key information shown on the SSH Access page.

Element	Description
Service Name	Filters the results to include SSH keys only for the specified deployment. You can enter a full or partial deployment name.
<b>Search</b>	Searches for SSH keys by applying the filters specified by the Service Name and Service Type fields, and displays the results in the table.
Results per page	Specifies the number of results you want to view per page. The default value is 10.
	Displays a description of an item in the results table. Clicking on the resulting downward arrow hides the description.
Service Name	Shows the name of the deployment.
Service Type	Shows the type of cloud service for this deployment.
Last Update	Shows the most recent time the SSH keys for this deployment were updated.
Actions	<p>Click the <b>Add New Key</b> button to add a new SSH public key to this deployment. The <b>Add New Key</b> overlay is displayed with its <b>Key value</b> field displaying the deployment's most recent SSH public key.</p> <p>Specify the new public key using one of the following methods:</p> <ul style="list-style-type: none"> <li>• Select <b>Upload a new SSH Public Key value</b> and click <b>Choose File</b> to select a file that contains the public key.</li> <li>• Select <b>Key value</b>. Delete the current key value and paste the new public key into the text area. Make sure the value does not contain line breaks or end with a line break.</li> </ul>

## MySQL Cloud Service IP Reservations Page

The MySQL Cloud Service IP Reservations page displays the IP reservations you have created. You use the page to view, manage, and create Oracle Compute Cloud Service security rules.

### What You Can Do From the MySQL Cloud Service IP Reservations Page

Use the IP Reservations page to perform the tasks described in the following topics:

- [Creating an IP Reservation](#)
- [Deleting an IP Reservation](#)

### What You See on the MySQL Cloud Service IP Reservations Page

The following table describes the key information shown on the MySQL Cloud Service IP Reservations page.

Element	Description
Create	Click to create a new IP reservation. Opens the Create IP Reservation dialog.
Search IP Reservations by name or region	Searches for IP Reservations.
IP Reservations	<p>List of all IP Reservations defined. Each IP reservation listing displays the following information:</p> <ul style="list-style-type: none"> <li>• Region—the name of the region the IP reservation belongs to.</li> <li>• Status—USED or UNUSED.</li> <li>• IP Addresses—the reserved IP addresses.</li> <li>• Created On—the date and time the IP reservation was created.</li> </ul> <p>To delete an IP Reservation, click the X icon.</p>

## MySQL Cloud Service Overview Page

The MySQL Cloud Service Overview page displays overview information for a MySQL Cloud Service database deployment.

### Topics

- [What You Can do from the MySQL Cloud Service Overview Page](#)
- [What You See on the MySQL Cloud Service Overview Page](#)


### What You Can do from the MySQL Cloud Service Overview Page


Use the MySQL Cloud Service Overview page to perform the tasks described in the following topics:

- [Viewing Detailed Information for a MySQL Server Deployment](#)
- [Scaling MySQL Cloud Service](#)

### What You See on the MySQL Cloud Service Overview Page

The following table describes the key information shown on the MySQL Cloud Service Overview page.

Element	Description
 (top of page, by service name)	<p><b>Menu</b> icon provides the following options:</p> <ul style="list-style-type: none"> <li>• <b>Start</b>—Starts the service, if stopped.</li> <li>• <b>Stop</b>—Stops the service, if running.</li> <li>• <b>Restart</b>—Restarts the service.</li> <li>• <b>Service Credentials</b>—Opens the <b>Update Service Credentials</b> dialog, enabling you to edit the Cloud Storage user name and password used to connect to the defined cloud storage container.</li> <li>• <b>Access Rules</b>—Opens the <b>Access Rules</b> page.</li> <li>• <b>SSH Access</b>—Opens the <b>Add New Key</b> dialog.</li> <li>• <b>Add Tags</b>—Opens the <b>Manage Tags</b> dialog, enabling you to add tags to your instance, or edit existing tags.</li> <li>• <b>Enterprise Monitor URL</b>—Opens the MySQL Enterprise Monitor in another browser tab.</li> <li>• <b>View Activity</b>—Opens the <b>Activity</b> page for the selected MySQL Cloud Service.</li> </ul>
Shortcuts	<p>The following shortcuts are available on the top-right of the <b>Overview</b> page:</p> <ul style="list-style-type: none"> <li>• Start Service—starts the service.</li> <li>• Stop Service—stops the service.</li> <li>• Restart Service—restarts the service.</li> <li>• Display Monitoring Information—populates the <b>Healthcheck results</b> frame of the MySQL Resources section.</li> </ul>
Summary	<p>Displays an overview of the service.</p> <ul style="list-style-type: none"> <li>• <b>Nodes</b>—Total number of compute nodes.</li> <li>• <b>OCPUs</b>—Total number of Oracle CPUs allocated across all deployments.</li> <li>• <b>Memory</b>—Total amount of compute node memory (RAM) allocated across all deployments.</li> <li>• <b>Storage</b>—Total amount of storage allocated across all deployments.</li> </ul>
Status	The current status of the service.
Version	Version of MySQL Server configured for the deployment. For example: 5.7.18.
MySQL Character Set	Name of the character set used by the MySQL Server.
MySQL Port	Port number used by the MySQL Server. Default is 3306.
Location	Cloud service location where this deployment is configured.
Backup Destination	Backup destination, if defined.
Backup Volume Location	Path to local backup location. Default is /u01/backup.
Cloud Storage Container	Name of the cloud storage container.

Element	Description
<b>MySQL Component</b>	<p>Lists information on the MySQL Server running on the compute node. The following information is available:</p> <ul style="list-style-type: none"> <li>• <b>State</b>—Current state of the MySQL Server.</li> <li>• <b>Connect Descriptor</b>—Connection string for the MySQL Server database.</li> <li>• <b>Compute Shape</b>—Compute shape of the compute node.</li> </ul>
<b>Resources</b>	<p>Lists information on the compute node. The following information is displayed:</p> <ul style="list-style-type: none"> <li>• <b>Host Name</b>—Full name of the compute node.</li> <li>• <b>OCPUs</b>—Number of OCPUs allocated to compute node.</li> <li>• <b>Public IP</b>—Public IP address of the compute node.</li> <li>• <b>Memory</b>—Amount of memory (RAM) allocated to the compute node.</li> <li>• <b>Instance</b>—Database type. If the <b>Display Monitoring Information</b> has run, a <b>Healthcheck</b> icon is displayed. Click to display performance details for the MySQL Server.</li> <li>• <b>Storage</b>—Total amount of storage allocated to the compute node.</li> <li>• <b>Healthcheck results as of ....</b>—Displays monitoring information on the compute node. Displayed only if the <b>Display Monitoring Information</b> process was performed.</li> </ul> <p> (for <b>Resources</b>)</p> <p><b>Menu</b> icon provides the following options:</p> <ul style="list-style-type: none"> <li>• <b>Restart</b>—Restarts the virtual machine.</li> <li>• <b>Scale Up/Down</b>—Opens the <b>Scale Up/Down</b> dialog for the deployment.</li> <li>• <b>Add Storage</b>—Opens the <b>Add Storage</b> dialog for the deployment.</li> </ul>

## MySQL Cloud Service Access Rules Page

The MySQL Cloud Service Access Rules page displays rules used to control network access to MySQL Cloud Service deployments. You use the page to view, manage, and create Oracle Compute Cloud Service security rules.

### Topics

- [What You Can Do From the MySQL Cloud Service Access Rules Page](#)
- [What You See on the MySQL Cloud Service Access Rules Page](#)




### What You Can Do From the MySQL Cloud Service Access Rules Page

Use the Access Rules page to perform the tasks described in the following topics:

- [Enabling Port Access by Enabling an Automatically Created Access Rule](#)
- [Enabling or Restricting Port Access by Creating an Access Rule](#)

### What You See on the MySQL Cloud Service Access Rules Page

The following table describes the key information shown on the MySQL Cloud Service Access Rules page.


Element	Description
Results per page	Specifies the number of results you want to view per page. The default value is 10.
Create Rule	Click to create a new rule. See <a href="#">Enabling or Restricting Port Access by Creating an Access Rule</a> .
Status	Displays an icon that indicates whether a rule is enabled or disabled.
	Indicates the rule is enabled.
	Indicates the rule is disabled.
Rule Name	Name of the rule. When creating a rule, this must start with a letter, followed by letters, numbers, hyphens, or underscores. The name cannot start with <code>sys_</code> .
Source	Hosts from which traffic is allowed. Possible values are <code>mysql_MASTER</code> , <code>PUBLIC-INTERNET</code> , or a custom value in the form of an IP address.
Destination	Security list to which traffic is allowed. This will be <code>mysql_MASTER</code> security list for the deployment.
Ports	Port or range of ports for the rule.
Description	Description of the rule (optional).
Rule Type	Type of rule. Rule types are: <ul style="list-style-type: none"> <li><code>DEFAULT</code>—Rules created automatically when the database deployment was created. Can be enabled or disabled, but not deleted. See <a href="#">Enabling Port Access by Enabling an Automatically Created Access Rule</a>.</li> <li><code>SYSTEM</code>—Rules created by the system. Cannot be enabled, disabled, or deleted.</li> <li><code>USER</code>—Rules created by you or another user. Can be enabled, disabled, or deleted.</li> </ul>
 (for rule)	Menu that provides the following options: <ul style="list-style-type: none"> <li><b>Enable</b>—Enable the rule.</li> <li><b>Disable</b>—Disable the rule.</li> <li><b>Delete</b>—Delete the rule (<code>USER</code> rules only).</li> </ul>

## MySQL Cloud Service Backup Page

You use the Backup page to manage backup and restore of a particular database deployment.

### What You See on the MySQL Cloud Service Backup Page

The following table describes the key information shown on the MySQL Cloud Service Backup page.


Element	Description
<b>Summary</b>	<p>Displays a summary of backup activities. The following information is displayed:</p> <ul style="list-style-type: none"> <li>• <b>Storage Cloud Volume Used</b>—Displays the amount of storage used by the backup in the configured cloud storage.</li> <li>• <b>Backup Volume Used</b>—Displays the amount of local storage used by the backup.</li> <li>• <b>Backup Volume Percent Used</b>— Displays the amount of local storage used as a percentage.</li> <li>• <b>Incremental Backups</b>—Displays the incremental backup schedule.</li> <li>• <b>Full Backups</b>—Displays the full backup schedule.</li> <li>• <b>Last Successful Backup</b>—Displays the date and time of the last successful backup.</li> </ul>
<b>Backup Now</b>	Click to create a full backup of the database deployment.
<b>Configure Backups</b>	Click to open the <b>Configure Backups</b> dialog. For more information, see <a href="#">Customizing the Current Backup Configuration</a>
<b>Restore Point In Time</b>	Click to open the <b>Restore Point In Time</b> dialog, which enables you to restore your database to a specific time within the backup record.
 (for each available backup)	Menu that provides the <b>Restore</b> and <b>Delete</b> options. Choose <b>Restore</b> to recover to the given backup. Choose <b>Delete</b> to delete the backup.
<b>Restore History</b>	Listing of restore operations on the database deployment. Click the triangle icon next to the title to view the listing.

## MySQL Cloud Service Patching Page

You use the Patching page to view available patches, initiate a patching process, and view details of the last patching process for a particular database deployment.

### What You See on the MySQL Cloud Service Patching Page

The following table describes the key information shown on the MySQL Cloud Service Patching page.

Element	Description
<b>Available Patches</b>	Lists the version of the patch, the release date, a link to the patch release notes, and whether the patch requires a restart..
<b>Description</b>  (for deployment)	<p>Description of the deployment.</p> <p><b>Menu</b> icon provides the following options:</p> <ul style="list-style-type: none"> <li>• <b>Precheck</b>—Performs validation checks on the existing deployment to ensure it is a valid patch.</li> <li>• <b>Patch</b>—Patches the deployment.</li> </ul>
<b>Patch and Rollback History</b>	Expand to see a description of the actions taken during the last patching operation.
<b>Rollback</b>	Click to roll back the last patching operation. For more information, see <a href="#">Rolling Back a Patch by Using the MySQL Cloud Service Console</a> .




## MySQL Cloud Service Snapshots Page

You use the Snapshots page to view available snapshots, initiate a snapshot process, and create cloned services from the available snapshots..

### What You See on the MySQL Cloud Service Snapshots Page

The following table describes the key information shown on the MySQL Cloud Service Snapshots page.

Element	Description
<b>Available Snapshots</b>	Lists the name of the snapshot, the description of the snapshot, and the creation time and date..
<b>Create</b>	Opens the Create Snapshot dialog.
 (for snapshot)	<b>Menu icon</b> provides the following options: <ul style="list-style-type: none"> <li>• <b>Create Clone</b>—Creates a new service, cloned from the selected snapshot.</li> <li>• <b>Delete</b>—Deletes the snapshot.</li> </ul>

## Create MySQL Cloud Service: Service Page

Service is the first page in the wizard you use to create a new database deployment, as described in [Creating a MySQL Server Deployment](#).

### What You See in the Navigation Area

Element	Description
<b>Cancel</b>	Click to cancel the Create Service wizard without creating a new database deployment.
<b>Next&gt;</b>	Click to advance to the Create Service: Service Details page.

### What You See in the Page Content Area

The following table describes the key information shown on the Create Service: Service page.

Element	Description
<b>Service Name</b>	The name for the new database deployment. The name: <ul style="list-style-type: none"> <li>• Must not exceed 50 characters.</li> <li>• Must start with a letter.</li> <li>• Must contain only letters, numbers, or hyphens.</li> <li>• Must not contain any other special characters.</li> <li>• Must be unique within the identity domain.</li> </ul>
<b>Description</b>	(Optional) A description for the new database deployment.
<b>Notification Email</b>	(Optional) An email address where you would like updates about the deployment-creation operation to be sent.

Element	Description
<b>Region</b>	(Available only if your identity domain is enabled for regions.) The region for the database deployment. If you choose a region that supports Oracle Cloud Infrastructure, the <b>Availability Domain</b> and <b>Subnet</b> fields are displayed, and the deployment will be created on Oracle Cloud Infrastructure. Otherwise, the deployment will be created on Oracle Cloud Infrastructure Classic. Choose <b>No Preference</b> to let MySQL Cloud Service choose an Oracle Cloud Infrastructure Classic region for you.
<b>Availability Domain</b>	(Available only on Oracle Cloud Infrastructure) The availability domain (within the region) where the database deployment will be placed.
<b>Subnet</b>	(Available only on Oracle Cloud Infrastructure) The subnet (within the availability domain) that will determine network access to the database deployment.
<b>IP Network</b>	(Not available on Oracle Cloud Infrastructure) (Available only if you have selected a region and you have defined one or more IP networks created in that region using Oracle Cloud Infrastructure Compute Classic.) Select the IP network where you want the database deployment placed. Choose <b>No Preference</b> to use the default shared network provided by Oracle Cloud Infrastructure Compute Classic. For more information about IP networks, see these topics in <i>Using Oracle Cloud Infrastructure Compute Classic</i> : <ul style="list-style-type: none"> <li>About IP Networks</li> <li>Creating an IP Network</li> </ul>
<b>Tags</b>	Metadata used to identify instances. Select the required tags from the drop-down list, or create the required tags. Tags can be defined as a comma-separated list of <b>keys</b> , or as a <b>key:value</b> pairs. For example: <b>productionDB,owner:John,admin:John</b> . You can search and filter on tags from the <b>Instances</b> page. You can also define tags on the <b>Instance Overview</b> page.

## Create MySQL Cloud Service: Service Details Page


Service Details is a page in the Create MySQL Cloud Service wizard you use to create a new database deployment. For more information, see [Creating a MySQL Server Deployment](#).

### What You See on the Service Details Page

The following table describes the key information shown on the Service Details page.

Element	Description
<b>Shape</b>	The number of Oracle Compute Units (OCPU) and amount of memory (RAM) for each compute node of the new database deployment. MySQL Cloud Service offers several OCPU/RAM combinations, as described in <a href="#">Computing Power</a> .

Element	Description
<b>SSH Public Key Edit</b>	<p>The SSH public key to be used for authentication when using an SSH client to connect to a compute node of the new deployment.</p> <p>Click <b>Edit</b> to specify the public key, either by uploading a file containing the public key value or by pasting in the value of a public key.</p> <p>If you paste in the value, make sure the value does not contain line breaks or end with a line break.</p>
<b>Usable Database Storage (GB)</b>	The amount of storage in GB for actual database data.
<b>Administration User</b>	The name of the MySQL Database Server administration user. Default is <code>root</code> .
<b>Administration Password Confirm Password</b>	<p>The password for the MySQL Database administrative users:</p> <p>The password:</p> <ul style="list-style-type: none"> <li>• Must be 8 to 30 characters in length.</li> <li>• Must contain at least one lowercase letter</li> <li>• Must contain at least one uppercase letter</li> <li>• Must contain at least one number</li> <li>• Must contain at least one of these symbols: <code>_</code> (underscore), <code>#</code> (hash sign), or <code>\$</code> (dollar sign).</li> </ul>
<b>Database Schema Name</b>	<p>The name for the database instance. This name:</p> <ul style="list-style-type: none"> <li>• Must not exceed 64 characters. Can contain letters, numbers and special characters (<code>_</code>, <code>\$</code>).</li> <li>• Must start with a letter.</li> <li>• Must contain only letters, numbers, or these symbols: <code>_</code> (underscore), <code>#</code> (hash sign), or <code>\$</code> (dollar sign).</li> </ul>
<b>Server Character Set</b>	The character set used by the server.
<b>Timezone</b>	Select a timezone for the MySQL server. The default is <code>SYSTEM</code> which sets the timezone of the MySQL server to the same value used by the compute node. See <a href="#">MySQL Cloud Service Supported Timezones</a> for a complete list.
<b>MySQL Port</b>	The port on which the server listens. The range of available ports is 3200 to 3399. The default is 3306.
<b>Configure Enterprise Monitor</b>	Controls whether MySQL Enterprise Monitor is installed. Select <b>Yes</b> to install, <b>No</b> to decline.
<b>Manager User</b>	User name of the MySQL Enterprise Monitor's manager user.
<b>Manager Password Confirm Manager Password</b>	<p>The password for the manager user.</p> <p>The password you enter:</p> <ul style="list-style-type: none"> <li>• Must be 8 to 30 characters in length.</li> <li>• Must contain at least one lowercase letter</li> <li>• Must contain at least one uppercase letter</li> <li>• Must contain at least one number</li> <li>• Must contain at least one of these symbols: <code>_</code> (underscore), <code>#</code> (hash sign), or <code>\$</code> (dollar sign).</li> </ul>

Element	Description
<b>Agent User</b>	Name of the Agent user. This user is used by the MySQL Enterprise Agents to connect to the MySQL Service Manager and relay the information collected about the host on which the agent is installed and any MySQL instances installed on that host.
	<div style="border-left: 2px solid #0070C0; border-right: 2px solid #0070C0; border-bottom: 2px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> <b>Note:</b></p> <p>No agent installations are provided at this time. It is currently only possible to install the MySQL Enterprise Service Manager, which can monitor any MySQL instances, local or remote, using a built-in agent. If you want to monitor MySQL instances installed on other MySQL Cloud Service instances, you can configure the MySQL Service Manager to monitor them, but cannot monitor a remote host's operating system, file system, or network interfaces. Although agent installations are not currently possible, it is mandatory to define an Agent User and Password to properly configure the MySQL Enterprise Monitor. These parameters can be changed later.</p> </div>
<b>Agent Password</b>	The password for the agent user.
<b>Confirm Agent Password</b>	The password you enter: <ul style="list-style-type: none"> <li>• Must be 8 to 30 characters in length.</li> <li>• Must contain at least one lowercase letter</li> <li>• Must contain at least one uppercase letter</li> <li>• Must contain at least one number</li> <li>• Must contain at least one of these symbols: _ (underscore), # (hash sign), or \$ (dollar sign).</li> </ul>
<b>Backup Destination</b>	Controls how backups for the deployment are to be configured: <ul style="list-style-type: none"> <li>• <b>Both Cloud Storage and Local Storage</b>—backups are configured to be created automatically and stored both on local storage and on an Oracle Storage Cloud Service container.</li> <li>• <b>None</b>—backups are not configured.</li> <li>• <b>Cloud Storage Only</b>—backups are configured to be created automatically and stored in an Oracle Storage Cloud Service container. No local backup is created for this option.</li> </ul> <p>For more information about backups and backup configurations, see <a href="#">About Backing Up Databases on MySQL Cloud Service</a>.</p>

Element	Description
<b>Cloud Storage Container</b>	<p>The URL where backups to cloud storage are to be stored. :</p> <ul style="list-style-type: none"> <li>For database deployments in Oracle Cloud Infrastructure, enter the URL of an existing Oracle Cloud Infrastructure Object Storage bucket. The URL is of the form:   <code>https://swiftobjectstorage.region.oraclecloud.com/v1/account/bucket</code></li> </ul> <p>For example:   <code>https://swiftobjectstorage.us-phoenix-1.oraclecloud.com/v1/mycompany/mybucket</code></p> <p>You must create this storage bucket before you begin creating the database deployment. See <a href="#">Object Storage API</a> in Oracle Cloud Infrastructure documentation.</p> <ul style="list-style-type: none"> <li>For database deployments in Oracle Cloud Infrastructure Classic, enter the URL of an Oracle Cloud Infrastructure Object Storage Classic container. The URL is of the general form:   <code>https://storage-instance-specifier/container</code></li> </ul> <p>The field is populated with the default URL of the storage associated with your ID domain and the container name <b>MySQLCS</b>. For example, assuming an ID domain named testDomain:   <code>https://testDomain.storage.oraclecloud.com/v1/Storage-testDomain/MySQLCS</code></p>


 **Note:**

If the account does not include an Object Storage service entitlement or if the region selected is an Oracle Cloud Infrastructure region, the container field is not autopopulated.

The **Create Cloud Storage Container** checkbox is enabled by default to create the container the first time a MySQL Cloud Service instance is created on the ID domain. To use a different name, change the *container* string to the desired container name.

 **Note:**

Do not use the storage container that you are using to back up MySQL Cloud Service deployments to cloud storage for any other purpose. For example, do not also use it to back up Oracle Java Cloud Service instances to cloud

Element	Description
	<div style="border: 1px solid #ccc; padding: 5px; background-color: #e6f2ff;">  storage. Using the container for multiple purposes can result in billing errors.         </div> <p>To determine the <i>storage-instance-specifier</i> value for your account, see About REST URLs for Oracle Storage Cloud Service Resources in <i>Using Oracle Cloud Infrastructure Object Storage Classic</i>.</p>
<b>Cloud Storage User Name</b>	<p>The user name of a user who has read/write access to the specified Oracle Storage Cloud Service container.</p> <ul style="list-style-type: none"> <li>For database deployments in Oracle Cloud Infrastructure, enter the user name you use to sign in to the Oracle Cloud Infrastructure console.</li> <li>For database deployments in Oracle Cloud Infrastructure Classic, enter the Oracle Cloud user name of the administrator of the Oracle Cloud Infrastructure Object Storage Classic container specified in <b>Cloud Storage Container</b>. Usually, this is your Oracle Cloud user name.</li> </ul>
<b>Cloud Storage Password</b>	<p>The password necessary to access the location specified in <b>Cloud Storage Container</b>.</p> <ul style="list-style-type: none"> <li>For database deployments in Oracle Cloud Infrastructure, enter the Auth Token generated in Oracle Cloud Infrastructure for the user that you specified. For more information, see <a href="#">Prerequisites for Oracle Platform Services on Oracle Cloud Infrastructure</a> in the Oracle Cloud Infrastructure documentation.</li> <li>For database deployments in Oracle Cloud Infrastructure Classic, enter the password of the Oracle Cloud user specified in <b>Cloud Storage User Name</b>.</li> </ul>
<b>Create Cloud Storage Container</b>	<p>(Not available on Oracle Cloud Infrastructure) Enabled by default. Create a new Oracle Cloud Infrastructure Object Storage Classic container as part of the database deployment creation, if the container does not exist already. Specify the container name and the Cloud Storage user name and password in the preceding fields.</p>
<b>Create Instance from Existing Backup</b>	<p>If Yes is selected, enables you to populate, or initialize, the new database with the data stored in an existing backup, in cloud storage.</p>
<b>Backup File Cloud Storage Path</b>	<p>The path to the backup file.</p>
<b>Username</b>	<p>The cloud storage user name.</p>
<b>Password</b>	<p>The cloud storage password.</p>
<b>Cancel</b>	<p>Click to cancel the wizard without creating a new database deployment.</p>
<b>Next</b>	<p>Click to advance to the Confirmation page.</p>

## Create MySQL Cloud Service: Supported Timezones

### MySQL Database Supported Timezones

This section lists the timezones supported by the MySQL database.

- (UTC-11:00) Samoa Standard Time(Pacific/Pago\_Pago)
- (UTC-10:00) Hawaii Standard Time(Pacific/Honolulu)
- (UTC-08:00) Pacific Standard Time(US/Pacific)
- (UTC-08:00) Pacific Standard Time(America/Tijuana)
- (UTC-07:00) Mountain Standard Time(America/Denver)
- (UTC-07:00) Mountain Standard Time(America/Phoenix)
- (UTC-07:00) Mountain Standard Time(America/Chihuahua)
- (UTC-06:00) Central Standard Time(US/Central)
- (UTC-06:00) Central Standard Time(Canada/Saskatchewan)
- (UTC-06:00) Central Standard Time(America/Matamoros)
- (UTC-06:00) Central Standard Time(America/Guatemala)
- (UTC-06:00) Central Standard Time(America/Monterrey)
- (UTC-05:00) Eastern Standard Time(US/Eastern)
- (UTC-05:00) Eastern Standard Time(US/East-Indiana)
- (UTC-05:00) Eastern Standard Time(America/Bogota)
- (UTC-04:00) Amazon Time(America/Manaus)
- (UTC-04:00) Chile Time(America/Santiago)
- (UTC-04:00) Venezuela Time(America/Caracas)
- (UTC-04:00) Amazon Time(America/Cuiaba)
- (UTC-04:00) Atlantic Standard Time(America/Halifax)
- (UTC-04:00) Paraguay Time(America/Asuncion)
- (UTC-03:30) Newfoundland Standard Time(Canada/Newfoundland)
- (UTC-03:00) Brasilia Time(America/Araguaina)
- (UTC-03:00) Brasilia Time(Brasil/East)
- (UTC-03:00) Uruguay Time(America/Montevideo)
- (UTC-03:00) Brasilia Time(America/Fortaleza)
- (UTC-01:00) Azores Time(Atlantic/Azores)
- (UTC) Greenwich Mean Time(Africa/Monrovia)
- (UTC) Greenwich Mean Time(Europe/Dublin)
- (UTC) Western European Time(Africa/Casablanca)
- (UTC) Coordinated Universal Time(UTC)
- (UTC+01:00) Western Africa Time(Africa/Windhoek)
- (UTC+01:00) Central European Time(Europe/Sarajevo)
- (UTC+01:00) Central European Time(Europe/Paris)
- (UTC+01:00) Central European Time(Europe/Amsterdam)
- (UTC+01:00) Central European Time(Europe/Prague)
- (UTC+02:00) Eastern European Time(Asia/Damascus)

- (UTC+02:00) Israel Standard Time(Asia/Jerusalem)
- (UTC+02:00) Eastern European Time(Asia/Beirut)
- (UTC+02:00) Eastern European Time(Africa/Tripoli)
- (UTC+02:00) Eastern European Time(Europe/Athens)
- (UTC+02:00) Eastern European Time(Europe/Kaliningrad)
- (UTC+02:00) Eastern European Time(Europe/Helsinki)
- (UTC+02:00) Eastern European Time(Asia/Amman)
- (UTC+02:00) Central Africa Time(Africa/Harare)
- (UTC+02:00) Eastern European Time(Africa/Cairo)
- (UTC+03:00) Eastern European Time(Europe/Istanbul)
- (UTC+03:00) Arabia Standard Time(Asia/Baghdad)
- (UTC+03:00) Moscow Standard Time(Europe/Moscow)
- (UTC+03:00) Arabia Standard Time(Europe/Riyadh)
- (UTC+03:00) Eastern Africa Time(Africa/Nairobi)
- (UTC+03:30) Iran Standard Time(Asia/Tehran)
- (UTC+04:00) Gulf Standard Time(Asia/Muscat)
- (UTC+04:00) Azerbaijan Time(Asia/Baku)
- (UTC+04:00) Armenia Time(Asia/Yerevan)
- (UTC+05:00) Turkmenistan Time(Asia/Ashgabat)
- (UTC+05:00) Pakistan Time(Asia/Karachi)
- (UTC+05:30) India Standard Time(Asia/Calcutta)
- (UTC+05:45) Nepal Time(Asia/Kathmandu)
- (UTC+06:00) Bangladesh Time(Asia/Dhaka)
- (UTC+07:00) Indochina Time(Asia/Bangkok)
- (UTC+07:00) Krasnoyarsk Time(Asia/Krasnoyarsk)
- (UTC+07:00) Novosibirsk Time(Asia/Novosibirsk)
- (UTC+08:00) Ulaanbaatar Time(Asia/Ulaanbaatar)
- (UTC+08:00) China Standard Time(Asia/Shanghai)
- (UTC+08:00) Australia Western Standard Time(Australia/Perth)
- (UTC+08:00) Irkutsk Time(Asia/Irkutsk)
- (UTC+08:00) Singapore Time(Asia/Singapore)
- (UTC+08:00) China Standard Time(Asia/Taipei)
- (UTC+09:00) China Standard Time(Asia/Yakutsk)
- (UTC+09:00) Japan Standard Time(Asia/Tokyo)
- (UTC+09:00) Korea Standard Time(Asia/Seoul)
- (UTC+09:30) Australia Central Standard Time(Australia/Adelaide)
- (UTC+09:30) Australia Central Standard Time(Australia/Darwin)



- (UTC+10:00) Australia Eastern Standard Time(Australia/Hobart)
- (UTC+10:00) Australia Eastern Standard Time(Australia/Brisbane)
- (UTC+10:00) Australia Eastern Standard Time(Australia/Sydney)
- (UTC+10:00) Vladivostok Time(Asia/Vladivostok)
- (UTC+10:00) Chamorro Standard Time(Pacific/Guam)
- (UTC+11:00) Magadan Time(Asia/Magadan)
- (UTC+12:00) New Zealand Standard Time(Pacific/Auckland)
- (UTC+12:00) Fiji Time(Pacific/Fiji)

## Create MySQL Cloud Service: Confirmation Page

Confirmation is the final page in the wizard you use to create a new database deployment. For more information, see [Creating a MySQL Server Deployment](#).

### What You See on the Confirmation Page

The Confirmation page presents a summary list of all the choices you made on the preceding pages of the wizard. In addition, it provides the controls described in the following table.

Element	Description
<Previous	Click to return to the Details page.
Cancel	Click to cancel the wizard without creating a new deployment.
Create>	Click to begin the process of creating MySQL Cloud Service deployment. The wizard closes and the MySQL Cloud Service console is displayed, showing the new deployment with a status of In progress.