



# MySQL database migration guide to Azure

Latest PaaS and IaaS deployments



This project was commissioned by Microsoft and AMD.

# Table of contents

<b>Introduction</b>	<b>1</b>	<b>iaaS deployments: MySQL on Azure iaaS</b>	<b>21</b>
PaaS and iaaS: What to know for MySQL deployments	1	<b>Planning for your MySQL deployment on Azure iaaS</b>	<b>21</b>
<b>PaaS deployments: Azure Database for MySQL – Flexible Server</b>	<b>3</b>	Azure VM series operating system choices	22
<b>Best practices for deploying Azure Database for MySQL – Flexible Server</b>	<b>3</b>	AMD processor-based VM series to consider	22
Understanding Azure Database for MySQL – Flexible Server	3	OS compatibility	22
Benefits of deploying Azure Database for MySQL – Flexible Server	3	Azure Disk Storage overview	22
Planning for your deployment on Azure PaaS	4	Azure Disk Storage types	22
Service tiers and Azure VM series overview	4	Azure Disk Storage features	23
Azure VM service tiers and workload considerations	4	Azure Elastic SAN	24
AMD processor-based VM series to consider	5	<b>Deploying MySQL on Azure iaaS</b>	<b>25</b>
Configurable options and limitations	5	Creating an Ubuntu VM	25
Storage considerations	6	Installing MySQL 8 on the Ubuntu VM	27
Backup considerations	6	Compute considerations	28
<b>Deploying Azure Database for MySQL – Flexible Server</b>	<b>7</b>	VM sizing	28
Creating an Azure Database for MySQL – Flexible Server via the Azure portal	7	CPU	28
Creating an Azure Database for MySQL – Flexible Server via Azure CLI	12	Memory	28
Overview of using Azure Resource Manager templates	13	Storage considerations	28
<b>Managing Azure Database for MySQL – Flexible Server</b>	<b>14</b>	Configuring multiple volumes	29
Scaling up an existing deployment	14	OS considerations	30
Managing Azure Database for MySQL – Flexible Server using the Azure portal	14	OS options	30
Connecting to Azure Database for MySQL – Flexible Server with Azure CLI and creating a database	15	Database considerations	30
Connecting to Azure Database for MySQL – Flexible Server with MySQL Workbench	15	MySQL parameters	30
HA best practices	16	Connection configuration parameters	30
Zonal (same zone) HA	17	InnoDB configuration parameters	30
Zone-redundant HA	17	Scale considerations	31
Security best practices	17	Scaling up	31
Data encryption	17	Scaling out	31
Regular updates/patching	17	HA considerations	31
User management and access controls	17	Using managed disks	31
Threat detection, prevention, and mitigation	18	Availability zones	31
Backup and disaster recovery best practices	18	Availability sets	32
Automated and on-demand backups	18	MySQL	32
Monitoring performance and health	18	Replication	33
Azure Monitor and alerts	18	Security considerations	33
Query performance insights	19	Azure confidential computing	33
Troubleshooting and Support	19	Identity management	34
Azure support plans	19	Azure Key Vault	34
Community resources	20	Azure Storage	34
Documentation and knowledge base	20	Disk security	34
		Azure Networking	34
		Azure Virtual Network	34
		Azure DDoS Protection	34
		Azure Firewall	35
		Azure Load Balancer	35
		MySQL security features and considerations	35
		Role-based access control (RBAC)	35
		Authentication methods	35

MySQL Keyring .....	35
Across the network .....	36
Encryption for data at rest .....	36
TCP/IP configuration .....	36
Firewall .....	36
Network segmentation .....	36
Encryption considerations .....	36
Backup considerations .....	37
Azure VM backup .....	37
MySQL backup considerations .....	37
Physical backups .....	37
Logical backups .....	37
Full and incremental backups .....	37
Full and incremental (a.k.a. point-in-time) recovery .....	38
Encrypting backups .....	38
Restoring backups .....	38
Monitoring and maintenance .....	38
Azure Monitor .....	38
Monitoring at the MySQL level .....	39
Other tools to consider .....	40
Percona MySQL .....	40
Charmed MySQL .....	40
<b>Conclusion</b> .....	<b>41</b>
<b>Appendix</b> .....	<b>46</b>
Sample ARM Template for deploying Azure Database for MySQL .....	46
template.json .....	46
parameters.json .....	50

## Introduction

MySQL™ is one of the most used open-source relational databases in the market, and organizations can easily pair it with Microsoft Azure technologies to gain the flexibility and reliability benefits of the cloud. With its beginnings in the 1990s,<sup>1</sup> MySQL has many features for handling structured and unstructured data, complex queries, and transactional or reporting applications. By choosing to run MySQL on Azure, customers can enjoy a myriad of VM choices, storage configurations, monitoring tools, security, and high availability (HA) options.

This document addresses best practices for deploying Azure Database for MySQL – Flexible Server and MySQL on Azure infrastructure as a service (IaaS). It covers VM selection, deployment procedures, monitoring and maintenance, storage considerations, database configuration considerations, and backup and restore.

## PaaS and IaaS: What to know for MySQL deployments

As organizations continue to shift their MySQL databases to the cloud, selecting the right deployment model becomes a critical decision that can impact flexibility, scalability, and operational efficiency. Two predominant cloud service models—Platform as a Service (PaaS) and Infrastructure as a Service (IaaS)—offer distinct advantages depending on the needs of an organization. PaaS streamlines application deployment by providing a ready-to-use platform that abstracts much of the underlying infrastructure complexity. IaaS provides a highly customizable cloud environment where businesses can have complete control over the operating system, storage, and networking.

By offering PaaS and IaaS solutions for MySQL, Azure allows organizations to select the model that best suits their needs while continuing to use familiar Microsoft and Azure tools. Choosing between the Azure PaaS and IaaS offerings comes down to the balance among organizations' needs for control, convenience, and available resources.

PaaS refers to cloud computing services that provide a managed platform for customers to develop, host, and manage applications without manually building, maintaining, and securing the underlying infrastructure. PaaS platforms deploy and configure infrastructure and automatically handle all maintenance, patching, and updates. With Azure Database for MySQL – Flexible Server, this means that customers can focus on business-critical activities, including application development, instead of devoting resources to database management tasks, such as HA, disaster recovery, backups, and geographic data replication.

PaaS offers these specific advantages:

- **Less coding time:** Pre-coded application components, such as workflows, directory services, security features, and search, can boost app creation.
- **Faster time to market:** Access to cloud capabilities, including development and devops tools, can help streamline the app dev process.
- **Easy scaling:** Scale up compute and storage to meet workload demands with the click of a button and near zero downtime while only paying for the resources you use.
- **Automated management and maintenance:** Azure handles HA, backups, patching, and provides intelligent tuning recommendations enabling organizations to reduce costs associated with administrative tasks while ensuring peak performance.

PaaS provides features that could bring additional advantages for staff and operations, including pre-coded application components, such as workflows, directory services, security features, and search.

In contrast to the benefits of PaaS, IaaS demands more technical resources and effort for deployment and configuration but provides complete control over the OS and database. This level of control is crucial in scenarios such as hosting legacy applications, meeting stringent regulatory and compliance requirements, or scenarios where organizations need higher levels of customization.

IaaS offers the following specific benefits:

- **Full control over the OS and database configuration:** For compliance and regulatory contexts (e.g., healthcare and finance), IaaS solutions might be valuable. In these use cases and others that require higher customization, admins will likely need more granular control over the database or data and security.
- **Flexibility to customize your configuration to meet your specifications:** Running MySQL on Azure IaaS allows customers to use whatever versions of MySQL they need while being able to configure all MySQL parameters. In contrast, PaaS allow customers to edit many, but not all, MySQL parameters. Additionally, legacy applications might depend on configurations or software versions that modern PaaS solutions do not support.

IaaS provides flexibility and a greater degree of control over your infrastructure. Manual deployment, management, and maintenance could require higher operational overhead and technical knowledge. For organizations that require this level of control and customization or that have in-house technical expertise, IaaS could be a good choice.

To make the right choice between PaaS and IaaS for your organization, you need to consider your specific needs and workloads. In this report, we explain how to plan and deploy either an Azure Database for MySQL – Flexible Server solution or MySQL on Azure IaaS.

# PaaS deployments: Azure Database for MySQL – Flexible Server

## Best practices for deploying Azure Database for MySQL – Flexible Server

The section covers deployment procedures and considerations for Azure Database for MySQL – Flexible Server, including compute selection, storage options, deployment methods, monitoring and maintenance considerations, failover and HA configurations, and backup and restore options.

### Understanding Azure Database for MySQL – Flexible Server

Azure Database for MySQL – Flexible Server is a fully managed, fully functional MySQL database PaaS platform. Azure automates database deployment, management, maintenance, hosting, and HA tasks for the virtual infrastructure hosting the database. After deployment, users and applications can access the database service via a client of their choosing. To manage shifting needs, admins can adjust database parameters to match workload requirements. They can also use the auto scale IOPS capability to cater to specific workload requirements and scale compute and storage resources dynamically in times of unexpectedly high demand.

Azure also offers robust networking, security, provisioning, and redundancy capabilities that can be costly and complex to maintain in a self-hosted environment.

Azure Database for MySQL – Flexible Server supports MySQL versions 5.7 and 8.0.<sup>2</sup> Azure automates regular patches and updates within the supported MySQL versions on either a customer-defined (where customers choose the day and time for Azure to perform maintenance) or system-managed maintenance schedule.<sup>3</sup> Azure Database for MySQL – Flexible Server is available in 55 Azure regions.<sup>4</sup>

### Benefits of deploying Azure Database for MySQL – Flexible Server

Deploying Azure Database for MySQL – Flexible Server gives you all the benefits that PaaS solutions in the Azure ecosystem offer—ease of deployment and configuration, on-demand scalability, simplified management and automated maintenance, resiliency and availability, and security. Azure Database for MySQL – Flexible Server also integrates with Azure App Service, Azure Kubernetes Service, and Azure Cognitive Services, which makes deploying and managing databases for cloud-native applications and workloads simple and convenient for developers working in the Azure ecosystem. Using a managed database service allows organizations to focus on development without devoting time to database management and maintenance, backups, manually scaling, or database security. To gain better scalability and performance, GeekWire decided to migrate its WordPress site to the Microsoft Azure platform. By taking advantage of fully managed services like Azure Database for MySQL, the company can scale on-demand while cutting costs 45 percent.<sup>5</sup>

Similarly, we previously conducted hands-on testing to demonstrate the ease and benefits of migrating an existing MySQL database to Azure Database for MySQL – Flexible Server on an AMD EPYC™ processor-based VM. Read the report at <https://facts.pt/WA4Av1Z>.

(Note: Azure Database for MySQL – Flexible Server is “a fully managed production-ready database service designed for more granular control and flexibility over database management functions and configuration settings.”<sup>6</sup> The offering has supplanted Azure Database for MySQL – Single Server, which Microsoft retired on September 16, 2024. Azure discontinued support for deploying Single Server instances from the Azure portal on January 16, 2023, and Azure CLI on March 19, 2024.<sup>7</sup>)

## Planning for your deployment on Azure PaaS

### Service tiers and Azure VM series overview

You can create an Azure Database for MySQL – Flexible Server instance in one of three service tiers, differentiated by the underlying Azure compute:

- B-series Burstable VMs
- D-series General Purpose VMs
- E-series Business Critical VMs

Within these service tiers, Azure offers multiple VM options.

Azure VMs come in predefined groupings, or series of configurations. Within each series type, there are generations (e.g., v5, v6), and each VM size within the series has a certain number of virtual CPUs (vCPUs) and amount of virtual RAM. Azure documentation recommends certain families of VMs for specific application use cases. For example, Microsoft recommends E-series and D-series VMs for relational database applications.<sup>8</sup> To explore the different VM series types and their common use cases, see <https://learn.microsoft.com/azure/virtual-machines/sizes>.

When deploying MySQL on Azure, you will determine the appropriate VM series and type for your specific use case. For production workloads, you will most likely select VMs from either the general purpose or memory optimized categories because MySQL will heavily use CPU, RAM, or both. You are unlikely to need high performance computing or GPU instances in a typical MySQL use case.

### Azure VM service tiers and workload considerations

Table 1 provides detailed information on the three service tiers.

Table 1: Azure Database for MySQL – Flexible Server VM sizing options. Source: Microsoft Azure.<sup>9</sup>

	VM type	Number of vCores	Processor OEM	Target workloads
<b>Burstable</b>				Web servers, development build environments, small databases
	Standard_B*ms	1, 2, 4, 8, 12, 16, 20	Intel®	
	Standard_B*s	2	Intel	
<b>General Purpose</b>				Enterprise applications, such as SAP or Microsoft Dynamics, high-traffic web servers, production-sized development environments
	Standard_D*ads_v5	2, 4, 8, 16, 32, 48, 64, 96	AMD	
	Standard_D*ds_v4	2, 4, 8, 16, 32, 48, 64	Intel	
<b>Business Critical</b>				Data analytics, large in-memory relational databases, scientific simulations, data warehousing
	Standard_E*ads_v5	2, 4, 8, 16, 20, 32, 48, 64, 96	AMD	
	Standard_E*ds_v5	2, 4, 8, 16, 20, 32, 48, 64, 96	Intel	
	Standard_E*ds_v4	2, 4, 8, 16, 20, 32, 48, 64	Intel	
	Standard_E*ids_v4	80	Intel	

B-series burstable VMs can be valuable for test or development workloads or use cases where baseline CPU utilization is low overall but may require periods of higher performance based on intermittent demand. In addition, burstable VMs do not offer HA.<sup>10</sup> Customers might also use B-series VMs for low-traffic web servers or applications, microservices, or other less critical workloads.

D-series general purpose compute VMs are the standard tier of Azure VMs. Customers might choose them for general production use cases and workloads. Microsoft recommends these VMs for enterprise customers with general purpose workloads that do not require a compute-to-memory ratio higher than 1:4 (for each vCore there is 4GiB memory). Examples of these workloads include e-commerce systems, production websites, hosted services, virtual desktop infrastructure hosting, and small to medium-sized databases.<sup>11</sup>

E-series business critical VMs “are optimized for in-memory applications”<sup>12</sup> and feature a higher than typical ratio of memory to compute power (up to 8GB of memory per vCore, compared to 4GB per vCore in the General Purpose VM service tier).<sup>13</sup> The Business Critical tier also offers accelerated logs, where logs reside on higher IOPS disks, at no additional cost.<sup>14</sup> Microsoft recommends these VMs for workloads that require in-memory processing such as analytics, high-performance transactional workloads, exceptionally large databases, or business intelligence applications.

### **AMD processor-based VM series to consider**

The two AMD EPYC™ processor-based VM series available for Azure Database for MySQL – Flexible Servers are the Dadsv5 General Purpose tier VMs and the Eadsv5 memory-optimized Business Critical tier VMs. The Dadsv5 series provides a balanced ratio of compute to memory resources with 4 GiB of memory per virtual core, while the Eadsv5 series VMs feature 8 GiB of memory per vCore in most configurations.<sup>15</sup>

The Dadsv5 series features scalability up to 96 vCPUs and 384 GiBs of RAM, and use AMD EPYC 7763v processors that “can achieve a boosted maximum frequency of 3.5GHz.”<sup>16</sup> For more information on the Dadsv5 VMs, see <https://learn.microsoft.com/azure/virtual-machines/dasv5-dadsv5-series>. Dadsv6 VMs based on 4<sup>th</sup> Gen AMD EPYC processors are currently in preview.<sup>17</sup>

Eadsv5-series VMs feature memory-heavy 8:1 RAM-to-vCPU ratios in most configurations,<sup>18</sup> with the smallest Eadsv5 VM having 16 GiBs of RAM with two vCores, and the largest VM configuration having 672 GiBs of RAM with 112 vCores. For more information on the Eadsv5 series, see <https://learn.microsoft.com/azure/virtual-machines/easv5-eadsv5-series>.

### **Configurable options and limitations**

Azure Database for MySQL – Flexible Server instances have several provisioning and configuration options to consider prior to deployment, including service region, availability zone, authentication method, access scope, and HA. In addition to choosing an Azure region for the server, administrators can specify a particular availability zone within that region to collocate the database service with other Azure services or hosted applications that will use it. Customers can provision HA in either zone-redundant or same-zone configurations during the initial server deployment or enabled after the fact in the Azure portal. See the section HA best practices for details.

Azure Database for MySQL – Flexible Server instances support both traditional MySQL authentication via a user-defined administrator username and password and Microsoft Entra authentication for accessing and administering the server. Customers can enable either or both depending on application and security requirements. Administrators can also control network access to the server in many granular ways. They can allow inbound connections from public IP addresses or whitelist specific IP addresses or ranges in the Azure firewall. They can also permit other Azure services to access the resource, or they can restrict access sources to specific private endpoints or Azure virtual networks. All Azure Database for MySQL – Flexible Server deployments support encrypted TLS/SSL connections.



Due to its nature as a PaaS offering, Azure Database for MySQL – Flexible Server does not offer the same level of operating system (OS) control as a self-managed IaaS deployment of MySQL. If the configurable options detailed in this deployment guide are not sufficient for your workload or use case infrastructure requirements, you might find that the higher level of OS control offered by an IaaS deployment works better for your needs, but at the cost of higher manual effort for backup, maintenance, security, scalability, and replication.

### Storage considerations

Azure Database for MySQL – Flexible Server sets maximum IOPS for each instance type, and the maximums differ for each instance size (visit <https://learn.microsoft.com/en-us/azure/mysql/flexible-server/concepts-service-tiers-storage> for detailed memory and sizing options):<sup>19</sup>

- Burstable: 5,000 IOPS
- General purpose: 20,000 IOPS
- Business critical: 80,000 IOPS

Administrators can set a pre-provisioned IOPS level during deployment or opt to let Azure auto-scale IOPS as needed. Administrators specify the initial storage size in GiB during deployment and can enable Azure to scale storage dynamically as needed to prevent interruption of service due to storage capacity. Administrators can also manually scale up the storage capacity at any time but cannot scale storage down after initially provisioning the server.

### Backup considerations

Administrators choose backup retention periods of 1 to 35 days for daily automated backups and can specify whether backups are locally redundant, zone-redundant (for zone-redundant HA deployments), or geographically redundant in paired Azure regions. Administrators can enable geo-redundancy at any time for locally redundant backups. In addition to daily automated server backup snapshots, Azure backs up transaction logs every five minutes.

Azure Database for MySQL – Flexible Server also allows administrators to initiate on-demand backups manually through the Azure portal. Azure can store up to 50 on-demand backups.<sup>20</sup>

## Deploying Azure Database for MySQL – Flexible Server

This section details the process and configuration options for deploying Azure Database for MySQL – Flexible Server via the Azure portal or the Azure CLI and discusses basic methods to connect to the MySQL resource once deployed. Our examples detail a specific configuration we chose for our sample deployment, but we discuss all relevant configuration options in the step-by-step instructions below.

### Creating an Azure Database for MySQL – Flexible Server via the Azure portal

1. **Navigate** to the Azure portal at <https://portal.azure.com/> and **log in** using your Azure credentials.
2. Using the search bar, **navigate** to the Azure Database for MySQL – Flexible Servers service **or select** it from the Azure services menu (see Figure 1).

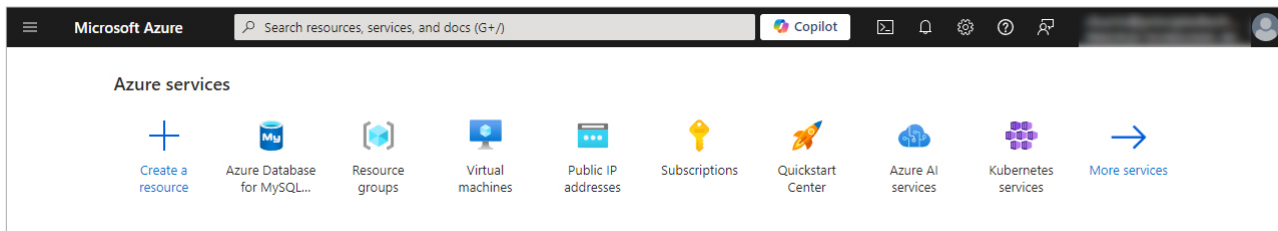


Figure 1: Screenshot of the Azure services menu after logging into the Azure portal. Source: Principled Technologies.

3. **Click Create**. Figure 2 shows the two locations of the Create button in the upper left and bottom center.

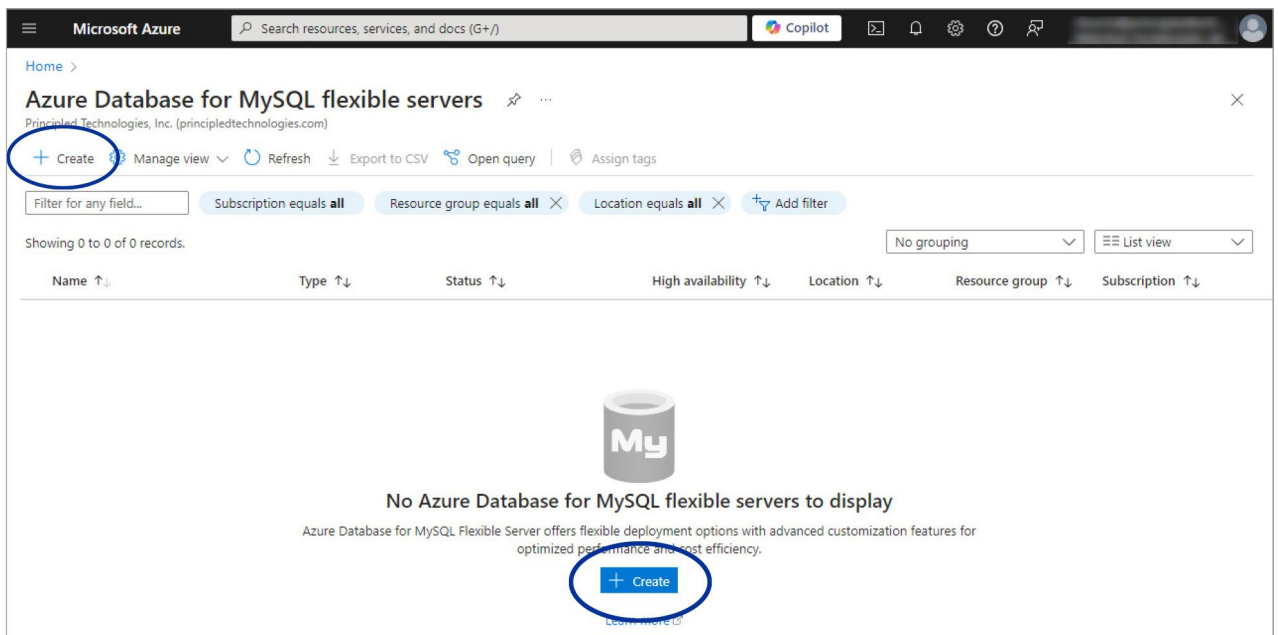


Figure 2: Screenshot of the Azure Database for MySQL – Flexible Server management page in the Azure portal. Source: Principled Technologies.

4. **Click** the Advanced Create button under the Flexible Server deployment option as shown in Figure 3.

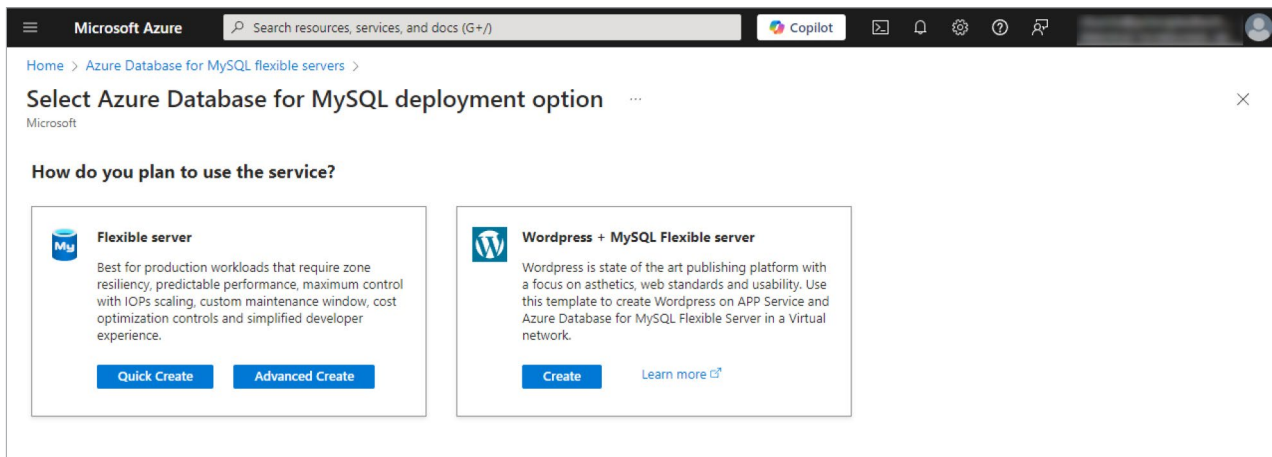


Figure 3: Screenshot of the Azure portal showing available Azure Database for MySQL – Flexible Server deployment options. Source: Principled Technologies.

5. Under Project Details, **select** the Azure subscription and resource group to which you want to deploy the resource. To create a resource group, **click** Create new under the dropdown menu or navigate to Resource groups via the search bar and **click** Create.

6. Under Server details:

a. **Enter or select** the following details for the server:

- **Server name** – a unique server name from 3 to 63 characters that may only contain lowercase letters, numbers, and hyphens; it cannot start or end with a hyphen
- **Region** – any of the Azure regions (for our sample deployment, we chose East US)
- **MySQL version** – either MySQL 5.7 (retired) or MySQL 8.0 (for our sample deployment, we chose MySQL 8.0)
- **Workload type** – This will determine which VM series and compute/storage resources the creation wizard recommends. Each of the three workload options corresponds to the three available Azure Database for MySQL – Flexible Server service tiers (General Purpose, Business Critical, and Burstable). For our sample deployment, we chose “For small or medium size databases.”
  - “For small or medium size databases” corresponds to the General Purpose compute tier and might be appropriate for most production and business workloads.
  - “Tier 1 Business Critical Workloads” corresponds to the Business Critical tier and might be appropriate for high-performance and real-time data workloads.
  - “For development or hobby projects” corresponds to the Burstable tier and might be appropriate for development or intermittent purposes where the server does not require full CPU usage all the time.

7. **Specify** the server compute and storage details:

a. **Click** Configure server to select compute and storage options. These options can give you more granular control over your VM series and compute tier. Figure 4 shows the Configure server page with our sample deployment values.

b. **Specify** the following options according to your workload needs:

- **Compute tier** – Choose among Burstable, General Purpose, or Business Critical.
  - For our sample deployment, we chose General Purpose.
- **Compute processor** – Choose the instance that is appropriate for your workload.
  - For our sample deployment, we chose an AMD EPYC™ instance. Note that only the General Purpose and Business Critical compute tiers offer AMD processors.
- **Compute size** – Choose one of the available VM series based on vCore, memory, and IOPS requirements.
  - For our sample deployment, we chose the Standard\_D16ads\_v5 VM series with 16 vCores, 64 GiB memory, and 20,000 max IOPS.
- **Storage size** – Specify the size of available storage capacity to make available to the server in gibibytes (GiB). This value can be anywhere between 20 GiB and 16 terabytes (TiB), and the solution uses the capacity for database files and logs. Customers can add storage after creating the server, but they cannot scale down this value creation.
  - For our sample deployment, we chose 64 GiB.
- **IOPS** – Select whether to auto-scale or pre-provision IOPS, which could generate more operating expenses.
  - For our sample deployment, we chose to auto-scale IOPS.
- **Accelerated logs (Business Critical tier only)** – Choose to enable Accelerated logs. Accelerated logs could enhance storage performance by optimizing log-related operations to improve throughput and cost efficiency. To learn more about accelerated logs, visit <https://learn.microsoft.com/azure/mysql/flexible-server/concepts-accelerated-logs>.
  - For our sample deployment, we enabled accelerated logs.
- **Storage auto-growth** – Choose to enable storage auto-growth, which could increase your billable storage capacity if the server runs out of storage capacity and prevents the server from becoming read-only and interrupting workloads.
  - For our sample deployment, we enabled storage auto-growth.
- **High availability** – Choose to enable HA. If customers enable HA, they can choose between same zone (zonal) HA or zone-redundant HA.
  - For our sample deployment, we enabled zone-redundant HA.
- **Backup options** – Select a backup retention period in days (1 to 35) and enable geo-redundancy. Note that users cannot enable geo-redundancy after server creation when using zone-redundant HA mode.
  - For our sample deployment, we specified a 14-day backup retention period and enabled geo-redundancy.

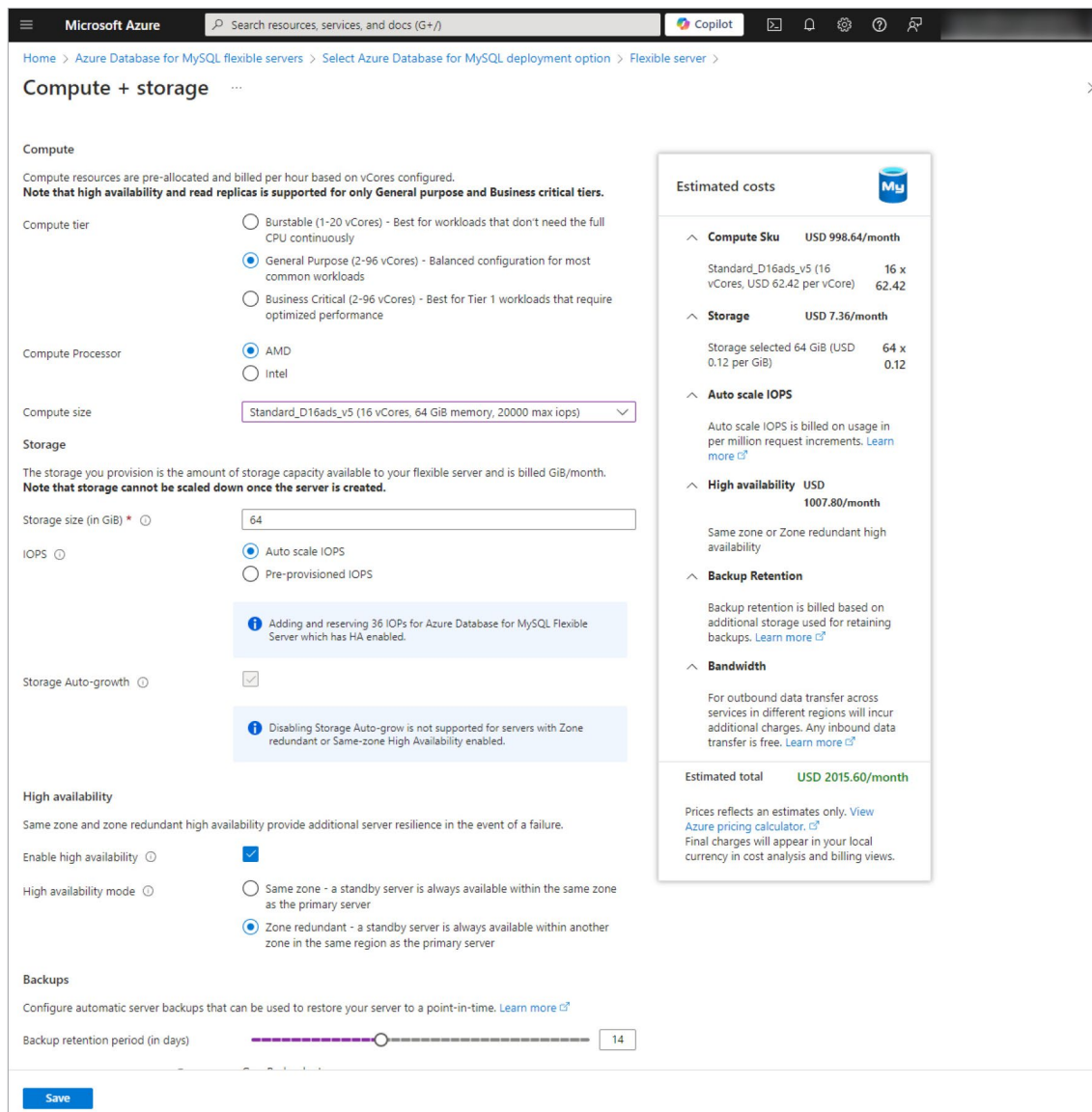


Figure 4: A screenshot of the Configure server page showing values for our sample deployment. Source: Principled Technologies.

- c. After configuring your compute and storage options, **click Save**.
- d. If desired, **select** a specific **Availability zone** for deployment. This might be useful if you want your database to exist in a specific availability zone that is co-located with an application or workload that uses it.
  - i. For our sample deployment, we chose no preference.
- e. **Choose Authentication Method:** MySQL authentication, Microsoft Entra authentication, or both.
  - i. For our sample deployment, we chose MySQL authentication.
- f. **Enter an Admin username and password** for the server. If you are using Microsoft Entra authentication, select a user-assigned managed identity and a Microsoft Entra admin.

## 8. Click Next → Networking.

- a. **Select connectivity method:** public or private access. Public access enables you to allow access to the server from all public IP addresses, to only other Azure services, to only a specific range of public IPs, through a private endpoint via an Azure virtual network, or any combination of the four. Private access limits access through using a private endpoint via Azure virtual network integration.
  - i. For our sample deployment, we chose public access.
- b. If using **public access**:
  - i. Choose to allow **public access** to this resource through the internet from a public IP address. For our sample deployment, we enabled this option.
  - ii. Choose to allow public access from any Azure service within Azure to this server. For our sample deployment, we enabled this option.
  - iii. Enter a **firewall rule** name and start and end IP addresses to give a range of public IP addresses access to the server if applicable. For our sample deployment, we added a rule to allow access to only our office public IP address.
  - iv. To allow private access, **create** a private endpoint by clicking **Add private endpoint** and specifying a virtual network, subnet, and private DNS zone.
- c. If using **private access** (VNet integration):
  - i. Select the Azure **subscription, virtual network, and subnet** to use for private access. Azure will create a virtual network and subnet in the same resource group by default if unspecified.
  - ii. Select a **private DNS** zone or leave blank to automatically create one.

## 9. Click Next → Security.

- a. Azure encrypts the storage by default using service-managed keys. If you would like to use a customer-managed key instead, select the user-assigned managed identity and key here.

## 10. Click Next → Tags.

- a. Enter any Azure tags based on organizational requirements that you wish to associate with the Azure Database for MySQL – Flexible Server.

11. **Click Next → Review** and create. This screen validates the server configuration and estimates the server creation time in minutes. Click Create to confirm and initiate server deployment. Figure 5 shows a successful Azure Database for MySQL - Flexible Server deployment using the Azure portal.

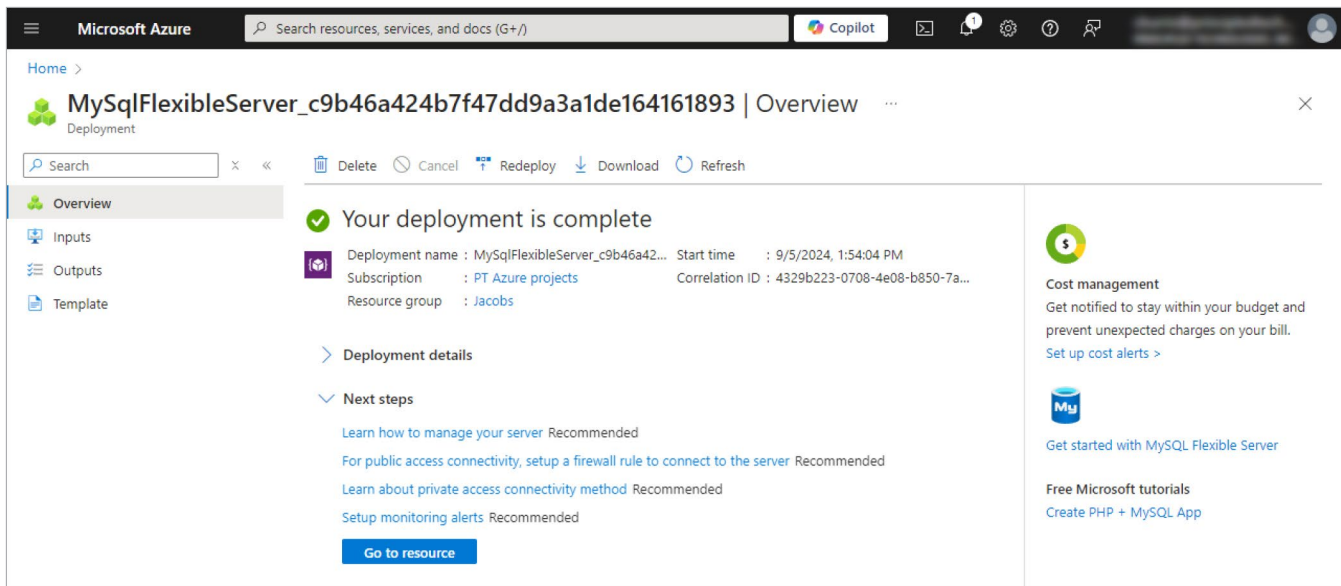


Figure 5: A screenshot of a successful Azure Database for MySQL deployment. Source: Principled Technologies.

## Creating an Azure Database for MySQL – Flexible Server via Azure CLI

Administrators can also deploy Azure Database for MySQL – Flexible Server directly from the command line using Azure Cloud Shell or Azure CLI. We used Azure Cloud Shell for our sample deployment, but the following steps should work with an authenticated user on Azure CLI version 2.0 or later as well.

1. **Open** Azure Cloud Shell by logging into the Azure portal and navigating to <https://portal.azure.com/#cloudshell/>. **Select** Bash when prompted.
2. **Select** your active Azure subscription if prompted, and click Apply.
3. If desired, **create** a new resource group for deployment:

```
az group create --name sampleresourcegroup --location eastus
```

4. **Create** a new Azure Database for MySQL server by running the following command and modifying the parameters and values as needed. You can find a list of all parameters for this command at <https://learn.microsoft.com/cli/azure/mysql/flexible-server?view=azure-cli-latest#az-mysql-flexible-server-create>. Any parameters without values will revert to Azure CLI defaults.

```
az mysql flexible-server create --name samplemysqlserver --resource-group sampleresourcegroup
```

5. **Respond** yes or no to any prompts that appear during the deployment process. Azure will restrict the server to private access by default, which will prompt you to allow your client IP address to connect. We enabled this in our sample deployment.

6. After the cloud shell console indicates that you have successfully deployed and configured the server, it will display a short JSON object output that shows the server connection string, associated identifiers and resources, the VM series SKU, and the generated admin username and password if you did not specify them in step 4. The server is now ready to use.
7. To output the server details and connection information in JSON format and save the output for reference, **run** the following command:

```
az mysql flexible-server show --resource-group sampleresourcegroup --name samplemysqlserver
```

8. **Run** the following command to test your connection to the database and confirm your client connects to the database server as expected, replacing <username> and <password> with the values from the JSON output in the previous step:

```
az mysql flexible-server connect -n samplemysqlserver -u <username> -p <password>
```

## Overview of using Azure Resource Manager templates

Azure Database for MySQL – Flexible Server supports provisioning via Azure Resource Manager (ARM) templates. An ARM template is a JSON file that defines the infrastructure and configuration for a deployment, which could consist of multiple databases on a single server or across multiple servers. ARM templates enable customers to automate provisioning and deployment of Azure Database for MySQL – Flexible Server instances with the same configuration options that users can specify when manually deploying instances via the Azure portal.

See [the appendix](#) for an example ARM template that you can use to deploy an Azure Database for MySQL – Flexible Server instance based on our sample configuration.



## Managing Azure Database for MySQL – Flexible Server

### Scaling up an existing deployment

Users can scale up or down Azure Database for MySQL – Flexible Server compute resources by adjusting the VM series type in the Azure portal. Users can also scale up storage, but not scale it down. Changing the VM type will require a server restart. You can follow these steps to scale up an existing Azure Database for MySQL – Flexible Server in the Azure portal:

1. **Log** into the Azure portal and navigate to the Azure Database for MySQL service using the services list or the search bar.
2. **Click** the name of the Azure Database for MySQL – Flexible Server instance to view its Overview page.
3. **Click** Settings in the left navigation pane to expand the settings, and click Compute + storage.
4. **Adjust** the compute tier, compute size (VM series) and storage size, IOPS, and auto-growth setting as needed. You can also enable post-deployment HA here or modify the backup retention period and geo-redundancy. The estimated costs panel at the bottom of the page reflects the new estimated monthly server cost based on configuration changes.
5. **Click** Save, and then when prompted to restart the server, click Continue.

### Managing Azure Database for MySQL – Flexible Server using the Azure portal

Users can manage Azure Database for MySQL – Flexible Server primarily using the Azure portal. To manage your server, select your Azure Database for MySQL – Flexible Server instance in the Azure portal and open the Overview panel by clicking the Overview menu. Then if you click Databases, you can click Add to create a new database on the server, which allows you to specify the database name, collation, and character set. You can also click Delete to remove a database from the server.

Administrators can also modify MySQL server parameters by clicking Server parameters in the left-side menu. There you can view all server parameters and modify them where possible. Modifying static parameters will prompt you to restart the server either immediately or later. This management option also allows you to reset all server parameters to default if they are misconfigured. For more detailed information about modifying MySQL server parameters, see <https://learn.microsoft.com/azure/mysql/flexible-server/how-to-configure-server-parameters-portal>.

The settings section of the menu also allows administrators to configure or enable replication, HA, backup settings, security settings, alerts and monitors, and other basic configuration options after server deployment.

## Connecting to Azure Database for MySQL – Flexible Server with Azure CLI and creating a database

1. **Open** the Azure Cloud Shell or Azure CLI, and connect to the server:

```
az mysql flexible-server connect -n samplemysqlserver -u  
<username> -p <password>
```

2. **Create** a new database, and specify the database name, the server name, and the resource group. If successful, you will see confirmation output consisting of database details in JSON format.

```
az mysql flexible-server db create -d sampledbname --server-name  
samplemysqlserver --resource-group sampleresourcegroup
```

3. **Confirm** deployment by connecting to the newly created database:

```
az mysql flexible-server connect -n samplemysqlserver -u <username> -p  
<password> -d sampledbname
```

4. **Run** the following command to test with a database query if desired:

```
az mysql flexible-server connect -n <server-name> -u <username> -p  
"<password>" -d <database-name> --querytext "<query text>"
```

For more information about the connect command, see the Azure CLI documentation: <https://learn.microsoft.com/azure/mysql/flexible-server/connect-azure-cli>.

## Connecting to Azure Database for MySQL – Flexible Server with MySQL Workbench

This section describes how to connect to a newly created Azure Database for MySQL – Flexible Server instance from the official MySQL workbench tool. This example assumes that you have allowed public access. If you have limited access to a private endpoint, then install MySQL workbench on another VM with access to the private virtual network instead of a client device.

1. **Download and install** the MySQL workbench application from the MySQL website <https://dev.mysql.com/downloads/workbench/>.
2. **Log into** the Azure portal and navigate to your Azure Database for MySQL resource. **Click** the resource to open the Overview panel.
3. **Note** the server hostname, admin username, and the password.
4. **Launch** MySQL Workbench.
5. **Click** the + to add a new database connection.
6. **Enter** a name for the connection, and **input** the server hostname, username, and password from step 3.
7. To verify the connection and credentials, **click** Test Connection, and **click** OK (see Figure 6).

8. **Click** the tile corresponding to the connection you just configured. After the client connects to the server, a new window will open to allow SQL query input.

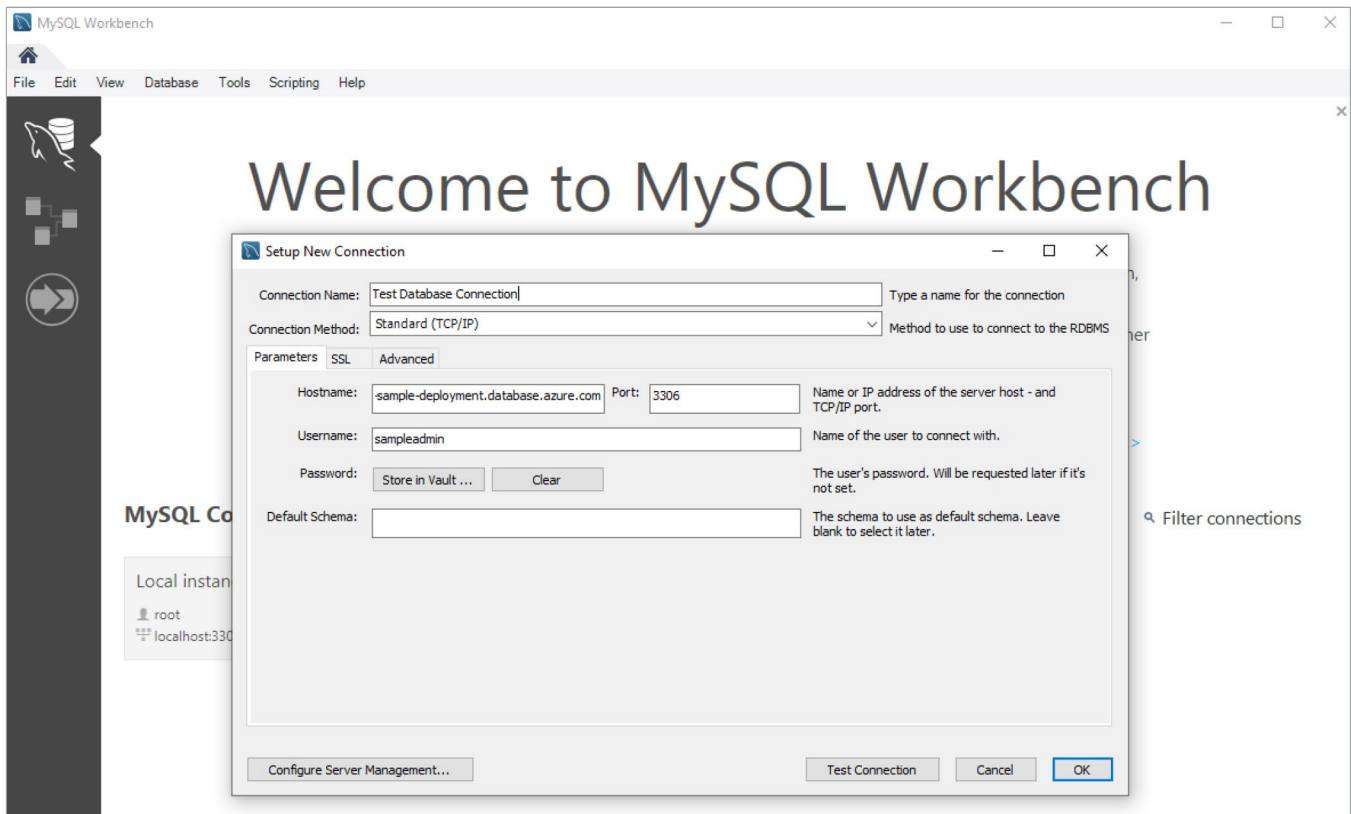


Figure 6: A screenshot showing the test connection window in the MySQL Workbench. Source: Principled Technologies.

For more information about using MySQL Workbench with Azure Database for MySQL – Flexible Server, visit <https://learn.microsoft.com/azure/mysql/flexible-server/connect-workbench>.

## HA best practices

Azure Database for MySQL – Flexible Server provisions physically separated primary and standby replicas in the same availability zone (zonal) or across availability zones (zone-redundant). Both configurations allow automatic failover with zero data loss.

Availability zones are separate data centers within the same region. Each data center has independent power, cooling, and networking. Azure services move to the nearest working availability zone in the event of a failure.

Azure always deploys standby replicas in the same VM configuration, and customers can remove them by disabling HA. Azure performs stop, start, and restart operations on both primary and standby servers at the same time.

Azure applies maintenance activities, such as updates, to the standby server first, and then promotes the standby server to primary while Azure updates the original primary server. This provides continuous uptime.

## Zonal (same zone) HA

Zonal redundancy (primary and standby replicas in the same availability zone) provides redundancy with lower latency than the zone-redundant configuration. Zonal redundancy could protect from node-level failures while reducing downtime during planned and unplanned events.

Azure backs up data and logs onto local redundant storage. Admins can choose the region and availability zone for deployment. Zonal redundancy offers an uptime SLA of 99.95 percent.<sup>21</sup>

## Zone-redundant HA

Zone redundancy (primary and standby replicas in different zones) provides the highest level of availability at the cost of some latency. The latency will show at the application level but is necessary to keep replicas up to date.<sup>22</sup> According to Azure, the latency will not affect read-only queries, but customers can expect a 5 to 10 percent impact for writes and commits.<sup>23</sup>

Zone redundancy provides complete physical separation between the primary and standby stacks. Azure mirrors data and logs three times onto zone-redundant storage for additional data protection.

Admins can choose the region and availability zones for both primary and standby locations. Zone-redundancy offers an uptime SLA of 99.99 percent.<sup>24</sup>

Note: The Burstable compute tier does not support HA.

## Security best practices

### Data encryption

Azure encrypts server, database, and backup data with a service-managed key by default, but customers can choose a user-managed key for encryption instead. Azure Database for MySQL – Flexible Server integrates with Microsoft Entra and Azure Key Vault to create an Azure-wide authentication method. Role-based access and user-managed identities give admins flexibility in managing data access. Customers can use customer-managed or service-managed keys to encrypt their Azure Database for MySQL – Flexible Server instances and backups. Azure stores the customer-managed key in an Azure Key Vault instance, which is highly available and scalable. Customers also have the option to have FIPS 140-validated hardware security modules back Azure Key Vaults. The Azure Key Vault and Azure Database for MySQL service must belong to the same Microsoft Entra tenant.<sup>25</sup>

### Regular updates/patching

Azure Database for MySQL – Flexible Server undergoes regular maintenance to ensure the managed database remains secure, stable, and current. During these maintenance periods, the server receives new features, updates, and patches. The standard maintenance cycle occurs at least every 30 days. Admins have the option to schedule maintenance on a specific day and time window of their choice, or they can allow the system to automatically select a day and time window for you. If you do not specify a preference, the system will choose a time between 11 PM and 7 AM in the server's regional time zone.<sup>26</sup>

### User management and access controls

MySQL authentication is the default method of connecting to Azure Database for MySQL – Flexible Server. However, admins can select Microsoft Entra as an additional or the sole method of connection.<sup>27</sup>

Admins can restrict network access via allow list (public IPs) or use Azure tools to create a private network for access control.<sup>28</sup>

## Threat detection, prevention, and mitigation

At an additional cost, admins can deploy Microsoft Defender for open-source relational databases to help monitor, investigate, and mitigate attacks. Microsoft Defender for open-source relational databases integrates with Microsoft Sentinel. After deploying Defender, the following activities will trigger alerts:

- Anomalous database access and query patterns
- Suspicious database activities
- Brute-force attacks

See the full list of alerts at <https://learn.microsoft.com/azure/defender-for-cloud/alerts-open-source-relational-databases>.

## Backup and disaster recovery best practices

### Automated and on-demand backups

Administrators choose backup retention periods of 1 to 35 days for daily automated backups and can specify whether backups are locally redundant, zone-redundant (for zone-redundant HA deployments), or geographically redundant in paired Azure regions. Users can enable geo-redundancy at any time for locally redundant backups. In addition to daily automated server backup snapshots, Azure takes transaction log backups every five minutes. If a daily backup fails for any reason, Azure retries the backup every 20 minutes until successful.

Azure Database for MySQL – Flexible Server also allows administrators to initiate on-demand backups manually through the Azure portal. Users can request and store up to 50 on-demand backups.<sup>29</sup>

## Monitoring performance and health

### Azure Monitor and alerts

Azure Monitor monitors Azure Database for MySQL – Flexible Server. All Azure metrics have a one-minute frequency with 30 days of history. Monitor pulls metrics, such as CPU and storage performance data, from Azure and the database. Admins can place metrics on a dashboard with various charts and alerts.<sup>30</sup> See the full list of metrics at <https://learn.microsoft.com/azure/mysql/flexible-server/concepts-monitoring>.

Some Azure Database for MySQL – Flexible Server metrics:

- Performance data
  - CPU, memory, and storage utilization percentages
  - Uptime
  - Credit usage (for burstable instance types)
- Storage breakdown
  - Capacity and limits
- Replication
  - Replication lag and status
- DML statistics
  - Select, update, insert, and delete statement counts
- DDL statistics
  - Create, drop, and alter db/table statement counts
- Innodb metrics
  - Lock and wait times
  - Buffer statistics

Admins can pull server logs using the Azure portal and CLI. Azure disables this feature by default, but admins can easily enable it in the Azure portal. Monitor retains logs for up to seven days or up to a maximum size of 7 GiB. The 7 GiB storage is free and non-extendable. Azure hashes out personally identifying information from logs, such as hostnames, IP addresses, and usernames. Admins can see this information by enabling audit logs.<sup>31</sup>

Because Azure Database for MySQL – Flexible Server integrates with Azure Monitor, admins can use Azure workbooks to parse and display metrics. Azure Database for MySQL – Flexible Server comes with three workbook templates: overview, auditing, and query performance. Admins can edit and customize templates as needed.

### Query performance insights

Admins can enable slow query logging (which Azure disables by default) to help troubleshoot slow queries. Admins can access the log via the Azure portal and CLI. Admins have access to several parameters to help filter the slow query logs:

- `long_query_time`
  - Determines the minimum length of logged slow queries
  - Default is 10 seconds
  - Does not affect active connections
  - Azure recommends restarting the database service after changing this parameter
- `log_slow_admin_statements`
  - Determines if administrative statements are logged
- `log_queries_not_using_indexes`
  - Determines if queries that do not use indexes are logged
- `log_throttle_queries_not_using_indexes`
  - Throttles the number of non-indexed queries that are logged

## Troubleshooting and Support

### Azure support plans

Azure offers four tiers of support plans. All customers have the basic Request Support tier, which includes access to Azure documentation and community resources, the ability to submit support tickets, access to the Azure Advisor service, and Azure health status notifications.

Azure offers the Developer support tier at \$29 per month. Azure intends for the tier to support test and development environments. The tier includes all of the resources in the basic tier with the addition of “third-party software support with interoperability and configuration guidance and troubleshooting” and offers full access to Azure support by email during business hours after customers initiate a support request.<sup>32</sup> Azure limits developer support tickets to severity level C (minimal business impact) and have a guaranteed response time of 8 business hours.

Next is the Standard support tier at \$100 per month. Azure intends for the tier to support production workloads and applications. It includes all resources and features of the basic and developer tiers with the addition of 24/7 access to technical support by phone or email. The tier also includes ticket creation with severity level B (moderate business impact) and level A (critical business impact) with response times of within 4 hours and 1 hour, respectively.<sup>33</sup>

Azure offers the Professional Direct support tier for \$1,000 per month. Azure intends for the tier to support business-critical workloads and applications. This tier includes all the features of the other tiers but with shorter response times: 4 business hours for severity level C, 2 hours (regardless of business hours) for severity level B, and 1 hour for severity level A. The tier also provides access to a pool of ProDirect delivery managers for architectural and operational support and service review and advisory consultation and guidance.<sup>34</sup>

For more detailed information about the Azure support plans available, visit <https://azure.microsoft.com/support/plans/>.

### Community resources

In addition to Azure support, there are several community resources available to supplement the official documentation and support resources and connect users with other Azure administrators and Microsoft engineers:

- **Azure community support** – Customers can find the landing page for the following resources at <https://azure.microsoft.com/support/community/>.
- **Azure on Q&A** (Microsoft community support forum) – Users can create a thread to ask questions about specific Azure services and get feedback or support from other community members and Microsoft experts at <https://learn.microsoft.com/answers/tags/133/azure>.
- **Azure on Stack Overflow** – Stack Overflow, a well-known third-party community resource for technical support, has an officially supported community channel at <https://stackoverflow.com/collectives/azure>.
- **Azure Support on Twitter** – Users can also log onto the social media site X (formerly known as Twitter) to engage with Azure experts at <https://x.com/AzureSupport>.

### Documentation and knowledge base

Azure Database for MySQL – Flexible Server has extensive official documentation, tutorials, and quick start guides for various common tasks and use cases:

- The official comprehensive documentation for Azure Database for MySQL – Flexible Server is available at <https://learn.microsoft.com/azure/mysql/>.
- Microsoft Learn includes a series of training tutorials for common use cases with Azure Database for MySQL – Flexible Server that users can access at <https://learn.microsoft.com/training/paths/introduction-to-azure-database-for-mysql/>.
- Azure offers a library of how-to technical training videos produced by Microsoft experts that cover various aspects of using Azure services and the Azure portal at <https://azure.microsoft.com/resources/videos>.

# IaaS deployments: MySQL on Azure IaaS

## Planning for your MySQL deployment on Azure IaaS

Whether on site or in the cloud, a MySQL deployment requires careful consideration of several factors, which we list below. Each environment and application is different, so this list is not exhaustive; you should consider your environment's specifics in your planning.

**Storage:** Azure offers several varieties of storage, each with a range of capabilities, that administrators can attach to virtual machines. Workloads that create a small-block random input/output (IO) pattern, such as most traditional online transaction processing (OLTP) workloads, may require virtual disks backed by high-speed SSDs that have higher input/output operations per second (IOPS) ratings. On the other hand, if your MySQL data is mostly at rest or does not need significant bursts of random IO, it may be best to consider more cost-effective, lower-IOPS options.

**VM selection and configuration:** Azure also provides many VM series with a range of specifications and costs. Azure has optimized each series for different use cases and workloads, so you can balance cost and performance according to your needs. You should carefully configure the VM you choose from the perspective of virtual storage and operating system software. Your choice of VM type will dictate most options, such as core count and virtual RAM, but you might consider configuring multiple volumes if you require a high degree of customization. For operating system choices, nearly any Linux distribution will work with MySQL. However, you'll likely want to use your organization's standard OS for compliance reasons, and you should check the MySQL release version's compatibility with the Linux® distribution version you choose at <https://www.mysql.com/support/supportedplatforms/database.html>.

**Database and VM sizing:** To select the appropriate Azure VM and disks, you will need some knowledge of your databases' sizes. You should consider the current sizing of all elements of the MySQL database application, such as tables, indexes, and log files. You should also consider their projected growth over time. We discuss this more in depth later in this document.

**MySQL configuration:** MySQL is an advanced database management system (DBMS), with many configuration options available that you can modify to affect your application performance, such as cache settings, IO concurrency, logging settings, checkpoint behaviors, and more.

**Networking:** Azure offers robust networking options. You'll want to be sure to explore the necessary networking options to set up the virtual networks you need, while controlling access to your MySQL database.

**Monitoring and maintenance:** While the PaaS offering from Microsoft includes patching and updates, the IaaS MySQL implementations are customer-supported and customer-patched. You should consider the maintenance needs of your MySQL implementation up front. In addition, you must monitor your database application, and Azure offers myriad options to do so. Combining these Azure Monitor tools with traditional MySQL monitoring approaches yields many options for a stronger implementation.

**Security:** Security is critical in any implementation, but especially in the cloud. In this section, we will review various security best practices for both Azure and MySQL.

**Backup and restore:** Any database implementation needs a carefully planned backup and restore framework. Later in this document, we discuss how to approach this critical piece of the puzzle in the context of running MySQL on Azure in an IaaS implementation.



## Azure VM series operating system choices

### AMD processor-based VM series to consider

There are three recommended AMD processor-based VM types: the Easv6 and Eadsv6 memory-optimized VMs and the Famsv6 compute-optimized VMs. All feature sizeable RAM-to-vCPU ratios, making them excellent candidates for database workloads where you need much of your working dataset cached in database buffers.

The Easv6 series and Eadsv6 series both feature scalability up to 96 vCPUs and 672 GiBs of RAM, and use AMD EPYC™ 9004 processors that "can achieve a boosted maximum frequency of 3.7GHz."<sup>35</sup> The Eadsv6 series VMs also include a local NVMe temp disk for application and system short-term storage, which is helpful for page files, swap files, and SQL tempdb files, among other things. For more information on the Easv6 VMs, see <https://learn.microsoft.com/azure/virtual-machines/easv6-eadsv6-series>. For more information on the Eadsv6 VMs, see <https://learn.microsoft.com/en-us/azure/virtual-machines/sizes/memory-optimized/eadsv6-series>.

Famsv6 VMs feature full cores (no simultaneous multithreading) and memory-heavy 8:1 RAM-to-vCPU ratios,<sup>36</sup> with the smallest Famsv6 VM having 16 GiBs of RAM, and the largest VM having 512 GiBs of RAM. For more information on the Famsv6 series, see <https://learn.microsoft.com/azure/virtual-machines/fasv6-falsv6-series>.

In addition to Famsv6 VMs, we also recommend Basv2, Falsv6 VMs, Fasv6, and Da-series v6 VMs for very small or non-production databases.

### OS compatibility

MySQL is compatible with many commonly available operating systems and is likely to run on all the operating systems the Azure catalog offers. MySQL documentation states that MySQL supports operating systems including Oracle Linux, Red Hat® Enterprise Linux®, CentOS, Ubuntu®, SUSE Enterprise Linux, Debian, Microsoft Windows and MacOS®.<sup>37</sup> For this guide, we chose Ubuntu as the underlying operating system.

## Azure Disk Storage overview

Because persistent and consistent data handling is a cornerstone of a reliable database application, storage design plays a key role in any database deployment. When designing in the context of Azure, you must consider Azure Disk Storage offerings and how to present those to the guest operating system, and thus, the database. In this section, we explore the Azure Disk Storage types and their features.

### Azure Disk Storage types

Azure Disk Storage is a service for managing Azure VMs' virtualized disk storage. Azure manages the underlying storage hardware, eliminating the need for users to maintain infrastructure. Users can simply choose a size and disk type, provision the disk, and then attach the disk to a VM.

Azure offers five disk storage types: Ultra Disk, Premium SSD v2, Premium SSD, Standard SSD, and Standard HDD. Table 1 shows the specifications for each as well as examples of workloads for which each disk type is suited. Note that while these limits apply specifically to Azure Disk Storage, Azure VMs also have their own unique storage performance limits depending on the VM series, as we discuss below.

Table 2: Comparison of storage performance maximums and workload scenarios for the disk types available via Azure Disk Storage.  
 Source: <https://learn.microsoft.com/azure/virtual-machines/disks-types>.

	Ultra Disk	Premium SSD v2	Premium SSD	Standard SSD	Standard HDD
Disk type	SSD	SSD	SSD	SSD	HDD
Scenario	IO-intensive workloads such as SAP HANA, top tier databases (for example, SQL, Oracle), and other transaction-heavy workloads.	Production and performance-sensitive workloads that consistently require low latency and high IOPS and throughput	Production and performance sensitive workloads	Web servers, lightly used enterprise applications and dev/test	Backup, non-critical, infrequent access
Max disk size	65,536 GiB	65,536 GiB	32,767 GiB	32,767 GiB	32,767 GiB
Max throughput	10,000 MB/s	1,200 MB/s	900 MB/s	750 MB/s	500 MB/s
Max IOPS	400,000	80,000	20,000	6,000	2,000, 3,000*
Usable as OS Disk?	No	No	Yes	Yes	Yes

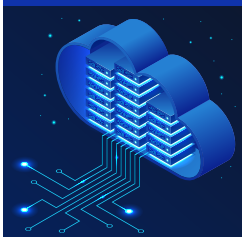
\*Applies only to disks with performance plus (preview) enabled.

### Azure Disk Storage features

These disks come equipped with several benefits and features.

**Storage availability and durability:** Azure has designed managed disks for at least 5 nines (99.99999 percent) of availability via high levels of durability and redundancy.<sup>38</sup> Azure achieves this redundancy by creating three replicas of your data, meaning that even in the event of one or two replicas failing, the remaining replicas can protect against data loss. In addition, locally redundant storage (LRS) disks provide at least 11 nines (99.9999999999 percent) of durability each year, and zone redundant storage (ZRS) disks provide at least 12 nines (99.999999999999 percent) of durability each year. Azure Disk Storage is integrated with both availability sets and availability zone features provided by Azure, providing further protection for user data. Availability set integration ensures that disks associated with VMs in an availability set are isolated from each other by placing the disks in different storage scale units known as stamps, thereby avoiding a single point of failure. Similarly, availability zone integration protects against a single point of failure by giving users the ability to place disks in data centers in separate physical locations.<sup>39</sup>

**Storage scalability and using multiple volumes:** Users can create up to 50,000 VM disks of a particular type per region in a subscription. Disks can range in capacity from 4 GiB to 64 TB, providing performance specs of anywhere from 120 to 400,000 IOPS and 25 MB/s to 10,000 MB/s of throughput, allowing you to customize the performance profile of your storage based on your needs.



In many cases, especially development or prototyping situations, it is possible to place data and log files on the same virtual disks. In more complex situations with larger databases or when optimizing for performance, we recommend splitting the data files, binary log, and redo log (assuming you are using the default InnoDB engine) onto separate volumes due to their differing IO types and to benefit from host caching. If you choose to use multiple volumes, you can configure these at instance creation or after the fact.

For the data drive, consider using Premium SSD v2 disks which provide additional features, high IOPS, and high throughput for a competitive price. For the log drive, plan for capacity projections. On all disk types, you should thoroughly test performance and evaluate costs. In general, the Premium SSD v2 disks or the Premium P30-P80 disks are good choices for high IOPS. If you require extremely low latency, also consider evaluating Azure Ultra Disks. Note that Azure offers Premium SSD v2 disks in certain regions only.

**Backup support for disks:** Managed disks include support for Azure Backup, a service that can create a backup job with scheduled backups and customizable retention policies. Additionally, Azure Backup includes Azure Disk Backup, a native solution that automates periodic creation of point-in-time snapshots for managed disks and retention of said backups based on user-defined policy. It supports backups for both OS and data disks, regardless of whether they are currently attached to a running Azure VM.<sup>40</sup> It is an agentless solution, meaning that the snapshot process does not affect application performance.

**Snapshots and images:** You can take a snapshot of your managed disks, creating a read-only full copy of the disk that exists independently of the source disk. You can use these to create new disks and to take a backup of a disk at any time. Azure takes snapshots from a single disk and stores them as standard managed disks by default. Azure incremental disk snapshots are a cost-effective way to create point-in-time backups of your Azure managed disks. Unlike full disk snapshots, which capture full disk data, incremental snapshots capture only the changes to disks since the last snapshot. This results in lower storage costs and faster snapshot creation times.<sup>41</sup>

Azure Disk Storage also supports the creation of custom images from either a user-managed VHD in a storage account or directly from a generalized VM. The capture process includes all managed disks attached to a VM, creating a single image. You can use these images to clone VMs that you have configured beyond the standard OS images provided by Azure. Azure Elastic SAN

### Azure Elastic SAN

Azure Elastic SAN is a service Azure provides that allows you to create and provision a cloud-based SAN interoperable with Azure VMs, as well as other Azure offerings such as Azure Kubernetes Service and Azure VMware Solutions.<sup>42</sup> Azure Elastic SAN is not limited to working with one particular type of storage-consuming host or service. Additionally, you can provision volumes from a single SAN to use across multiple different Azure offerings. Azure simplifies the process of creating the SAN by handling the deployment, management, and initial configuration.

Elastic SANs can provide up to millions of IOPS with double-digit GiB/s of throughput and single-digit-millisecond latencies.<sup>43</sup> Its benefit in a MySQL use case would be if you had to store and centrally manage many terabytes of database data across multiple MySQL instances. However, storage latencies on Elastic SAN would likely be higher than those of Premium SSD v2 instances or Azure Ultra Disks.

The individual volumes that you provision from the SAN share the SAN's performance values, up to a limit of 80,000 IOPS per volume. Individual volumes can have high IOPS and throughput, but the total IOPS and throughput of all your volumes can't exceed the total IOPS and throughput your SAN has. In addition to these performance benefits, Azure Elastic SAN brings a centralized approach to management and volume creation and maintenance, like in more traditional on-premises storage array approaches. You should consider Azure Elastic SAN as your storage solution when you need highly scalable storage with centralized management for large-scale applications or when your workloads require low latency, high IOPS, and high throughput across multiple volumes.<sup>44</sup>

Because you can connect Azure Elastic SAN with your Azure Virtual Machines, you could choose to back your MySQL databases with Azure Elastic SAN storage if your organization requires that approach.

## Deploying MySQL on Azure IaaS

### Creating an Ubuntu VM

Deploying MySQL on an Azure VM involves creating the necessary VMs and configuring Azure specifics—such as selecting subscriptions, regions, and disks—and then accessing the VM's operating system and performing the necessary MySQL installation steps.

To demonstrate this, we used Ubuntu version 24.04 and a standard E2ads\_v6 VM. We created an Ubuntu virtual machine by logging into the Azure Portal, navigating to the Virtual Machines service, and clicking Add. We set the Azure options and chose our OS image, Ubuntu 24.04 LTS. Next, we configured access essentials such as generating a new key pair and opening port 22. Finally, we configured disk and networking options, reviewed our VM, and clicked Create.

For step-by-step instructions, see below.

1. Log into the Azure Portal, and navigate to the Virtual Machines service.
2. To open the Create a Virtual Machine wizard, click Create → Azure Virtual Machine.
3. On the Basics tab, set the following:
  - a. Choose your Subscription from the drop-down menu.
  - b. Choose your Resource group from the drop-down menu.
  - c. Name the Virtual Machine.
  - d. Choose your Region from the drop-down menu. (We used East US.)
  - e. Under Availability options, choose Availability zone.
  - f. Choose your Availability zone. (We used Zone 3.)
  - g. From the Image drop-down menu, choose Ubuntu 24.04 LTS.
  - h. Under VM architecture, select x64.
  - i. Leave Azure Spot VM set to No.
  - j. Select the VM size you wish to use. (We used Standard E2ads\_v6.)
  - k. Keep the Authentication type set to SSH public key.
  - l. Choose a Username and Key pair name, and select Generate a new key pair.
  - m. Leave Public inbound ports set to Allow selected ports.
  - n. For Select inbound ports, choose SSH (22).
4. Click Next: Disks.
5. On the Disks tab, set the following:
  - a. For the OS disk size, select 30GB.
  - b. For the OS disk type, choose Premium SSD from the drop-down menu.
  - c. Leave the default Encryption type.
6. Click Next: Networking.
7. On the Networking tab, leave all settings as default, and ensure the checkbox for accelerated networking is enabled.
8. Click Next: Management. On the Management tab, leave all defaults.

9. On the Advanced tab, leave all defaults. On the Tags tab, add any tags you wish to use.

10. On the Review + create tab, review your settings, and click Create.

After Azure has successfully deployed the VM, you can check its status in the Azure portal (see Figure 7).

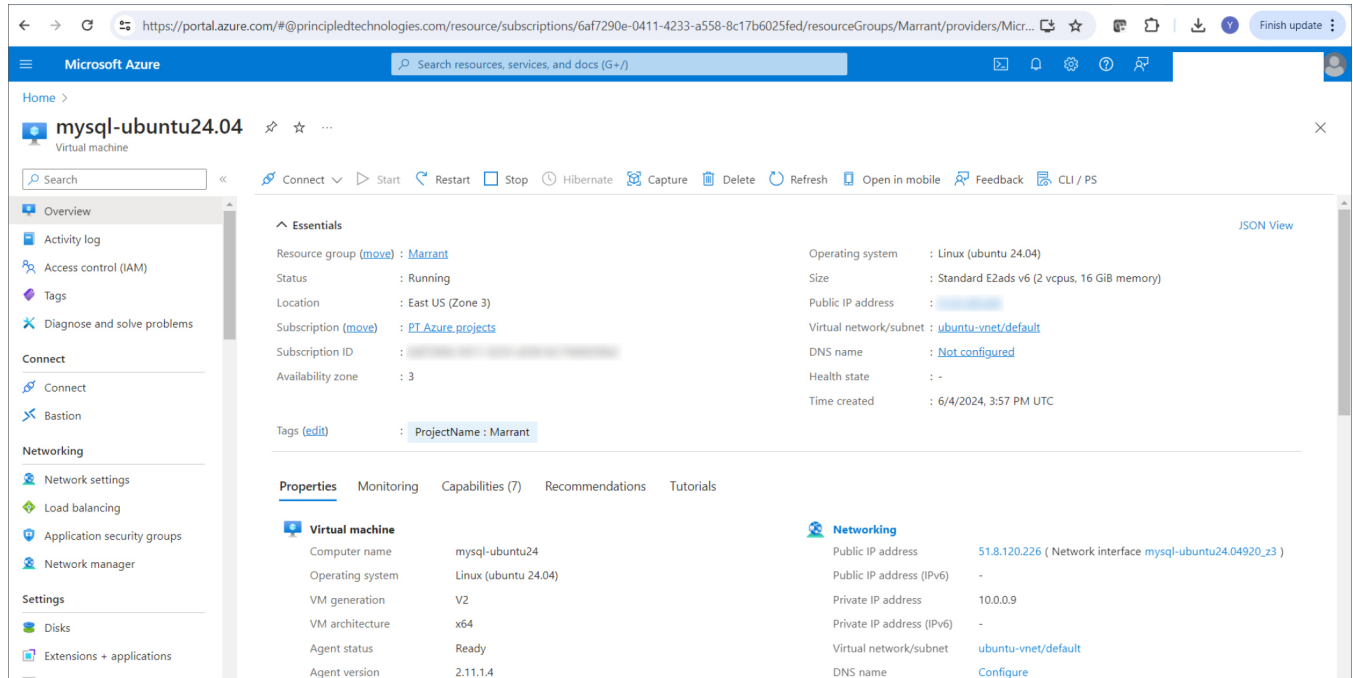


Figure 7: Screenshot of the Azure portal. Source: Principled Technologies.

## Installing MySQL 8 on the Ubuntu VM

In this section, we present the steps to install MySQL on the VM. Note: Consider using Ubuntu Pro (<https://ubuntu.com/pro>), which includes Expanded Security Maintenance (ESM), security updates, and other tools.

1. Use SSH to connect to the Ubuntu VM.
2. Update the repository:

```
sudo apt update
```

3. Install and start MySQL 8:

```
sudo apt install mysql-server-8.0
sudo systemctl start mysql.service
```

4. By default, the root account is not configured with a password. Set the root password:

```
sudo mysql
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password
BY '<password>';
mysql> exit
```

5. After installation, run MySQL security script to change default settings and enhance overall security:

```
sudo mysql_secure_installation
```

6. Follow the prompts to set a password, remove anonymous users and test database, etc.:
  - a. Enter y to set up a valid password.
  - b. Enter y to remove anonymous users.
  - c. Enter y to disallow root login remotely.
  - d. Enter y to remove test database and access to it.
  - e. Enter y to reload the privilege tables.
7. Once MySQL secure installation is complete, connect to the database:

```
sudo -u root -p
```

## Compute considerations

### VM sizing

VM sizing for MySQL involves optimizing Azure resources such as vCPUs, memory, and storage to improve database performance, scalability, and reliability. Below, we look at some key components in a VM and how they could impact performance.

### CPU

MySQL can benefit from multi-vCPU VMs on Azure for parallel query processing. You need to monitor CPU usage closely to make sure your MySQL server works efficiently for complex operations such as joins, sorting, aggregation, etc. You can use the built-in tools such as `PERFORMANCE_SCHEMA` to monitor CPU usage and troubleshoot time-consuming queries. Additionally, you can use native Linux tools such as `top`, `iostat`, and `nmon` to monitor your server's CPU usage as well.

### Memory

Memory is crucial for databases to function properly. During a typical database query, if the data is not already in memory, the database retrieves it from disk and then temporarily stores it in memory. Then, the database takes further query actions, such as any aggregation, grouping, and sorting. This makes it important that your VM has enough memory.

MySQL implements various cache mechanisms such as the query cache,<sup>45</sup> InnoDB buffer pool, and key cache<sup>46</sup> to store frequently accessed data in memory and minimize disk I/O latency. Additionally, MySQL benefits from Linux OS filesystem caching, even though it does not directly manage these caches. It is important to tune both MySQL and the Linux OS caches to maximize the advantages of caching and optimize the use of system resources.

## Storage considerations

Choose storage that meets the performance requirements of your application. Organizations should first measure the application demands, then use those IOPS measurements and projections to size and select Azure disks. Based on those projections, customers can choose both the disk layout (i.e., using multiple disks), and the disk model. A good starting point can be to use Premium SSD v2 due to its flexibility with adjusting IO performance. If you do select Premium SSD (non v2), then you could consider striping multiple less expensive disks if you require more IOPS than a single larger more expensive disk could deliver.

When choosing Azure Disk Storage for your high-performance applications, it is important to consider the disk's IOPS, throughput, and latency characteristics to ensure they meet your application's needs. OLTP applications need to process many concurrent user requests immediately. These requests are typically small and random IO operations that require very high disk IOPS. On the other hand, OLAP or data warehouse applications need to scan and read large volumes of data. These operations are generally large and sequential IO tasks that require high disk throughput.<sup>47</sup>

You can also consider configuring multiple disks into a RAID group to combine their IOPS, throughput, and storage capacity to deliver even higher performance for your applications. On Linux, you can use the `MDADM` utility to stripe disks together and create a software RAID.<sup>48</sup>

Choose an appropriate file system type (e.g., `ext4` or `XFS`) and mount option (e.g., `noatime`, `nobarrier`, or `nodiratime`) for MySQL data directories to optimize disk IO performance and reduce file system overhead. Consider using separate file systems or mount points for MySQL data, binary logs, redo logs and other general logs<sup>49</sup> to minimize contention and improve IO throughput (see the "Configuring multiple volumes" section below).

In addition to the IOPS and throughput limits on individual managed disks, users must also consider the performance limits of the VM itself. Each Azure VM has its own defined limit for IOPS and throughput, which, combined with the disk-specific limits, define the overall performance limits for the solution. Many VM series include limits for both cached and uncached disk performance, and burstable VMs have a specific limit for performance while bursting.



For example, say you were using a Standard\_E16ads\_v5 VM with an attached 256GiB Ultra Disk with 60,000 IOPS and 10,000 MB/s of provisioned throughput. Even though Azure has designed the disk to provide the given specifications, it will throttle the storage performance due to the inherent limits of the VM itself, which supports a maximum uncached IOPS of 25,600 and a maximum uncached throughput of 600 MB/s. Therefore, users must take both sets of limits into account, as well as the disk caching and bursting abilities of both the managed disks and the VMs. To find the storage performance limits of a particular VM or VM series, users can navigate to the [“Sizes for virtual machines in Azure”<sup>50</sup>](#) page, and select a VM type and VM series. Each series has an associated table listing the storage performance limits, which users can leverage to help plan their deployments.

### Configuring multiple volumes

Historically, in specific cases, database architects and storage architects have recommended configuring multiple volumes with multiple underlying disks for database applications. This is not always necessary, and you should evaluate your situation as to whether this approach would provide performance or reliability advantages.

Examples of when to do this may include:

- **Improved performance by separating data and logs:** The IO profile that data volumes ingest (small block, random, read/write) is likely different from the IO profile of a log volume (potentially larger block, sequential, write only).
- **Better scalability:** To increase the throughput available to you, you could stripe virtual disks together using the `lvcreate` tool to increase the IOPS capacities that are available to you in Azure on a single volume. Note that you should keep your organization's data redundancy policies in mind.
- **Streamlined backup and restore:** MySQL DBMS might benefit from separating data and log volumes when it comes to backup and restore speeds.
- **Different disk types for data and logs:** Because Azure offers a variety of disk choices, you can select the disk type that suits your performance needs specific to both the data and log storage of your MySQL database.



## OS considerations

### OS options

Adjust kernel parameters (sysctl settings) related to memory management, networking, disk IO, and process scheduling to optimize system performance for MySQL. Common kernel parameters to tune include `vm.swappiness`, `vm.dirty_ratio`, `vm.dirty_background_ratio`, `fs.file-max`, `net.core.somaxconn`, and `kernel.shmmax`. You can also consider various Linux filesystems, I/O schedulers, and NUMA settings. For a detailed explanation, please refer to <https://www.percona.com/blog/linux-os-tuning-for-mysql-database-performance/>.

## Database considerations

### MySQL parameters

MySQL has many parameters to optimize its performance. For an exhaustive list of all server parameters please visit <https://dev.mysql.com/doc/refman/8.0/en/optimization.html>. We discuss some important ones in this section.

### Connection configuration parameters

Table 3: Connection configuration parameters. Source: Principled Technologies.

Parameter	Description
<code>max_connections</code>	Sets the maximum number of concurrent connections to the MySQL server. Setting an appropriate <code>max_connections</code> value based on your application's requirements and server resources aims to prevent resource contention at connection.

### InnoDB configuration parameters

InnoDB is the default storage engine in MySQL. It aims to provide high performance, reliability, and concurrency for transactional workloads. Below are several key parameters for improving performance:

Table 4: InnoDB configuration parameters. Source: Principled Technologies.

Parameter	Description
<code>innodb_buffer_pool_size</code>	Defines the size of InnoDB buffer pool, which caches frequently accessed data and indexes. Increasing this parameter can improve read performance by caching more data in the memory and reducing disk read latency for frequently accessed data.
<code>innodb_log_file_size</code>	Sets the size of InnoDB log files, which log all database changes. Adjusting <code>innodb_log_file_size</code> can impact write performance.
<code>innodb_io_capacity</code>	Specifies the IOPS capacity for InnoDB background tasks such as flushing and checkpointing. You can adjust <code>innodb_io_capacity</code> to optimize I/O performance based on your underlying storage solutions.
<code>innodb_flush_method</code>	Defines the method used to flush in-memory data to InnoDB data files. This parameter can affect your I/O throughput. You can choose the appropriate flush method to enhance performance based on your operating systems and storage configurations.
<code>innodb_use_native_aio</code>	Controls whether InnoDB uses native asynchronous I/O. On Linux systems that support native asynchronous I/O, enabling this parameter can lead to better I/O performance for InnoDB.
<code>innodb_read_ahead_threshold</code>	Defines the number of sequential pages that InnoDB uses to prefetch to the buffer. This parameter can help improve performance for workloads with more sequential access patterns.
<code>innodb_random_read_ahead</code>	Determines whether random read ahead is enabled. This parameter can help improve performance for workloads that involve more random reads.

## Scale considerations

### Scaling up

As your business grows and you begin to see workloads increase and your resource utilizations approaching their limits, you might want to consider scaling up your MySQL VM to increase resources such as CPU, memory, and storage to handle more work.<sup>51</sup> You can change your VM size by using either the Azure Portal or Azure CLI. Scaling up will require you to first shut down the VM, which causes downtime. You should plan accordingly to minimize the impact of server outage. Scaling up to larger VM sizes will also incur higher costs. Please visit the Azure Linux Virtual Machines pricing calculator at <https://azure.microsoft.com/pricing/details/virtual-machines/linux/> and make sure you understand the cost implications.

### Scaling out

Another option to accommodate increased workloads is to scale out your MySQL VM. You can use Azure Virtual Machine Scale Sets (VMSS) to deploy and manage a set of identical and auto-scaling VMs. VMSS can automatically adjust the number of VMs in the scale set based on a predefined rule by scaling out VMs when workload increases and scaling in VMs when workload decreases. You can create VMSS from the Azure Portal or using Azure CLI. You can set the maximum and minimum instance count and scaling rules based on CPU, memory, or disk usage.<sup>52</sup> For example, you can add rules to automatically spin up a new VM in the scale set when your CPU usage exceeds a certain threshold. The ideal use case for this would be if you needed separate, smaller, isolated MySQL instances.

## HA considerations

Azure offers a variety of HA solutions to protect your applications and data from data center outages and maintenance events.<sup>53</sup>

### Using managed disks

As we note in the Azure disk storage features section, Azure managed disks are replicated three times and offer 3 nines of availability (99.999 percent) and 11 nines of durability (99.999999999 percent).<sup>54</sup> Because MySQL is an application that would typically run on a single VM, using the Ultra Disks, Premium SSD v2, or Premium SSD offerings from Azure would work well these types are all managed.<sup>55</sup>

### Availability zones

Availability zones are physically separate locations within an Azure region. Each zone has its own power, cooling, and networking. Deploying your MySQL VMs across multiple availability zones can enhance resiliency against data center failures.<sup>56</sup> If one zone is compromised, your replicated apps and data are still available in another zone. Availability zones offer uptime percentages with Azure SLA of 99.99 percent.<sup>57</sup>

Azure has designed scale sets to support large-scale services that require HA, auto-scaling, and easy management of a group of VMs. You can deploy virtual machines in a VMSS into multiple availability zones or fault domains. If one of these VM instances has a problem, customers can continue to access your MySQL application through one of the other VM instances with minimal interruption.<sup>58</sup>

## Availability sets

You can deploy your MySQL VMs on Availability sets, which distribute VMs across multiple fault and update domains. An update domain is a group of VMs and underlying physical hardware that you can update or reboot at the same time. A fault domain is a group of VMs that share common power source and network switch. By spreading VMs across multiple fault and update domains, Azure ensures that your application remains available even if one domain is being updated or failed. Availability sets meet the 99.95% Azure SLA.<sup>59</sup>

Microsoft recommends that users who are not already using Availability sets for their HA strategy deploy MySQL VMs across availability zones using VMSS flexible orchestration mode. They recommend this approach because VMSS flexible orchestration mode offers higher Azure SLA, larger VM scale, and enhanced VM management capabilities.<sup>60</sup>

## MySQL

MySQL provides many features that help customers build applications, efficiently manage data, protect data integrity, and provide fault-tolerant and HA environments.

In this section, we focus on some key concepts and methods related to MySQL HA and replication. For a full list of MySQL features, please visit <https://dev.mysql.com/doc/refman/8.3/en/>.

MySQL HA ensures a MySQL database system always remains accessible and operational even in the event of failures or disruptions. Key components of MySQL HA include replication, clustering, failover mechanisms and load balancing. Failover occurs when the primary (source) server becomes unavailable due to network issues, hardware failures, software crashes, or planned maintenance, and a standby server (replica) takes over as the new primary server. MySQL HA solutions such as MySQL Group Replication and MySQL InnoDB Cluster provide automated failover mechanisms, built-in monitoring, and management tools to facilitate seamless failover operations and maintain continuous database availability in production environments.<sup>61</sup>



It is important to consider Azure infrastructure design when designing your MySQL HA setup. Azure groups its resources into Regions, which contain zones. One could choose to deploy an HA configuration for MySQL both in a same-zone configuration, where both primary and secondary VMs reside within the same Azure zone, or across zones.

If you choose to deploy within the same zone, deploy VMs using the Azure Availability Set features to ensure VMs reside in separate fault domains.

If you chose to deploy across zones to ensure issues in one zone do not affect the uptime of your application, configure the primary VM in Zone 1, configure the secondary VM in zone 2, and then configure replication options.

## Replication

MySQL supports copying data from the source database server to one or more replica servers. Organizations commonly use replication for data redundancy, scalability, HA, and enhanced performance. MySQL replication supports different types of synchronization. By default, replication is asynchronous, meaning that the source server does not wait for the confirmation from the replica servers. In a MySQL group replication cluster or a MySQL Network Database (NDB) cluster, replication is synchronous, ensuring that transactions are received on all nodes before acknowledging to the client.<sup>62</sup> MySQL now also supports delayed synchronization where a replica deliberately lags behind the source by at least a specified amount of time. Delayed synchronization can be useful to protect against user mistakes on the source and test how the system behaves when there is a lag caused by heavy load on the replica.<sup>63</sup>

There are two primary types of replication formats in MySQL: statement-based replication (SBR) and row-based replication (RBR). SBR replicates complete SQL statements from the source to the replica, and RBR replicates only the modified rows. Additionally, MySQL allows you to configure a hybrid approach that combines both statement-based and row-based replication methods, choosing the most suitable method based on the nature of the changes to be replicated.<sup>64</sup>

## Security considerations

When you deploy your MySQL server on Azure IaaS, you can leverage numerous security features that both Azure and MySQL provide to help protect your data, applications, and infrastructure.

### Azure confidential computing

Azure confidential computing is a set of technologies and features in Azure that work to protect data at rest and in transit. These technologies include a number of different Azure offerings, including confidential VMs such as ECadsv5/ECasv5 powered by AMD EPYC™ CPUs. Microsoft lists the following features of confidential VMs:<sup>65</sup>

- “Encryption for ‘data in use,’ including the processor state and the virtual machine’s memory.
- Host attestation helps you verify the full health and compliance of the server before data processing begins.
- Hardware Security Module (HSM) can be attached to guard the keys of confidential VM disks, which the tenant exclusively owns.
- New UEFI boot architecture supporting the guest OS for enhanced security settings and capabilities.
- A dedicated virtual Trusted Platform Module (TPM) certifies the health of the VM, provides hardened key management, and supports use cases such as BitLocker.”



Only certain families of VMs offer Azure confidential computing, so you should check Microsoft documentation to ensure your relevant VM family is included. The ECasv5, ECadsv5<sup>66</sup>, DCasv5, and DCadsv5<sup>67</sup> series VMs all offer Azure confidential computing.

## Identity management

### Azure Key Vault

Azure Key Vault is a cloud-based service Azure provides that allows you to securely store and manage confidential information such as keys, secrets, and certificates. Azure stores the information in a secure and centralized vault and protects it from unauthorized access. Azure Key Vaults can be either software-protected or hardware-protected by HSMs. Microsoft documentation states that “software-protected keys, secrets, and certificates are safeguarded by Azure, using industry-standard algorithms and key lengths. For situations where you require added assurance, you can import or generate keys in HSMs that never leave the HSM boundary.”<sup>68</sup> When deploying a MySQL server in Azure, you can store your sensitive information, such as your passwords, certificates, and database connection strings, in a secure Azure key vault to enhance security for your database server and applications.

### Azure Storage

#### Disk security

Azure provides two important security features for its managed disks: encryption and Private Links. Two different types of encryption are available: server-side encryption and Azure Disk Encryption. Server-side encryption, which Azure enables by default for all managed disks and their associated snapshots and images, provides encryption at rest. The encrypting keys can be managed either by Azure (platform-managed keys) or by the user (customer-managed keys). Azure Disk Encryption uses OS-level features to encrypt OS and data disks used by Azure VMs, such as BitLocker for Windows-based VMs<sup>69</sup> or DM-Crypt for Linux-based VMs.<sup>70</sup> The process integrates with Azure Key Vault, allowing users to manage the disk encryption keys and secrets.

Private Links let you import or export a managed disk within a secure Azure virtual network without exposing traffic to the public internet. This service allows users to generate a time-bound Shared Access Signature (SAS) uniform resource identifier (URI) for managed disks that enables them to export the disk data to other regions.

### Azure Networking

#### Azure Virtual Network

Azure Virtual Network (vNET) allows you to create private, isolated, and secure network environments in the Azure cloud. Each vNET is logically isolated from other vNETs. You can define IP address ranges, subnets, and network security groups (NSGs) to control traffic flow and enforce security policies between your MySQL server VMs and applications within vNETs. To connect with other vNETs, Virtual Network Peering establishes peering connections between vNETs to enable seamless communication and resource sharing. To provide outbound internet connectivity, the Azure Network Address Translation (NAT) gateway can route VM traffic to the internet while keeping VMs and compute resources private. If you are deploying a containerized MySQL environment, vNET provides a container network interface (CNI) plugin to enable communication and connectivity for your containers within your Azure Kubernetes Service cluster.<sup>71</sup>

#### Azure DDoS Protection

“A distributed denial-of-service (DDoS) attack is a malicious attempt to disrupt the normal traffic of a targeted server, service, or network by overwhelming the target or its surrounding infrastructure with a flood of Internet traffic.”<sup>72</sup> It is crucial to implement proactive security measures to protect against and mitigate the impact of DDoS attacks. Azure DDoS Protection automatically detects and mitigates even the most complex DDoS attacks against Azure resources, including virtual networks, public IP addresses, and Azure Load Balancers. It provides always-on monitoring and automatic mitigation capabilities to defend against both the network layer and the application layer attacks.<sup>73</sup>

## Azure Firewall

Azure Firewall allows you to define network security policies and rules to control inbound and outbound traffic for your Azure virtual network resources, based on IP addresses, ports, protocols, etc. You can create allow/deny rules, application rules, network rules, and NAT rules. Azure Firewall integrates with Azure Monitor for viewing and analyzing firewall logs.<sup>74</sup>

## Azure Load Balancer

Azure Load Balancer distributes inbound network traffic across backend pool VMs such as VMs and VM scale sets, based on configured load-balancing rules and health probes. It helps evenly distribute network workload, prevent bottlenecks, and optimize network resource utilizations. Azure Load Balancing supports both public load balancing and internal load balancing. A public load balancer can provide a public IP address for virtual machines inside your vNET. Public load balancers load balance internet traffic to your VMs. An internal or private load balancer handles traffic inside a virtual network or across vNET peering connections.<sup>75</sup>

## MySQL security features and considerations

### Role-based access control (RBAC)

MySQL RBAC allows you to define and manage user access permissions based on roles. Roles are sets of privileges that admins can grant to users or other roles. Permissions are sets of operations (e.g., SELECT, INSERT, UPDATE, DELETE, CREATE, and DROP) that a role can perform on database objects. Admins can grant privileges to roles and revoke them. They can grant roles to a user account, which gives to the account the privileges associated with each role. With RBAC, you can implement fine-grained access controls, manage permissions through roles, and enhance the security and integrity of your MySQL databases.<sup>76</sup>

### Authentication methods

MySQL supports several authentication methods to verify user credentials and grant access to databases. By default, MySQL uses the `mysql_native_password` method, which hashes user passwords and stores them in the `mysql.user` table in the MySQL system database. `sha256_password` and `auth_socket` are more secure methods: `sha256_password` hashes passwords using the SHA-256 algorithm, and `auth_socket` plugin allows users to authenticate with MySQL server using the operating system's credentials.<sup>77</sup>

Additionally, MySQL Enterprise Edition supports more authentication capabilities through LDAP, and pluggable authentication module (PAM), and Kerberos plugins. With LDAP authentication, MySQL stores user credentials in an LDAP directory, and validates these credentials against the LDAP server.<sup>78</sup> In PAM authentication, MySQL delegates the authentication process to an external PAM unit that uses various methods supported by PAM.<sup>79</sup> The Kerberos plugin enables users to authenticate to the MySQL server using Kerberos tickets.<sup>80</sup>

### MySQL Keyring

The MySQL Keyring component and plugin provide secure storage and management of sensitive data. Keyring enhances security by protecting these sensitive items from unauthorized access and provides a centralized mechanism for managing encryption keys and credentials. Keyring component can store keys in either a regular file locally on the host (`component_keyring_file`), in an encrypted and password-protected file on the host (`component_keyring_encrypted_file`), or even inside a vault in cloud (`component_keyring_oci`). Similarly, the keyring plugin stores keys in either a local file or an encrypted local file. The keyring service routines are a set of loadable functions that developers can call at both the SQL level and C language level.



### Encryption considerations

For data security, consider all stages of the data lifecycle: data at rest on the cloud, in transit, and back to cloud storage. A good list of best practices could be:

1. Always use disk encryption.
2. Encrypt data in transit.
3. Use SSL host certificates.
4. Consider targeted encryption at the row or cell level in MySQL.

### Across the network

MySQL supports Secure Sockets Layer (SSL) and Transport Layer Security (TLS) encryption between clients and servers. SSL/TLS encryption secures client-server communication and protects data in transit. SSL connections encrypt all data that travel across the network, including passwords, queries, and returned data. You can configure MySQL server to use SSL/TLS by specifying the paths to the SSL certificate, key, and CA certificate in the MySQL server configuration file `my.cnf`.<sup>81</sup>

### Encryption for data at rest

MySQL InnoDB storage engine provides data-at-rest encryption and ensures that data stored on disk is encrypted and protected from unauthorized access. You can encrypt individual tables or tablespaces to make sure that data stored in the encrypted tables is protected. MySQL leverages keyring plugins to manage encryption keys securely. These keyring plugins store encryption keys local to the database server but outside the database to enhance security.<sup>82</sup>

### TCP/IP configuration

MySQL uses TCP/IP as the default communication protocol for client-server connections. The MySQL server listens for incoming TCP/IP connections on a specified port (default is 3306) and IP address.<sup>83</sup> The `/etc/mysql/mysql.conf.d/mysqld.cnf` file contains TCP/IP-related settings, including the listening address, port number, and socket.

### Firewall

A best practice is to deploy a firewall on the MySQL database VM. Linux distributions offer built-in firewalls such as `ufw`, `firewalld`, and `iptables`. A firewall enables you to set rules that govern allowed inbound and outbound traffic. You can define these rules based on parameters such as the local port (default is 3306 for MySQL), protocol (IPv6 or TCP), and source address (often a list of subnets or specific addresses). Strictly controlling external access to the server is crucial for maintaining the security of your database.

### Network segmentation

Users can utilize network segmentation to separate MySQL servers from other network traffic and restrict access to authorized users and applications. Within Azure, you can leverage Azure Virtual Network to create a private network specifically for your MySQL server and associated applications.

## Backup considerations

There are several options for backing up your MySQL VM on Azure. You can back it up at Azure level or use MySQL utilities to back up the database natively.

### Azure VM backup

When setting up an Azure VM for backup, a background backup process initiates based on your backup schedule.<sup>84</sup> To perform application or file system backups, you need to install a backup extension within the VM. The backup process starts by taking a snapshot of your VM and then transferring it to the vault. The first backup is a full backup, meaning Azure transfers all data blocks to the vault. In subsequent backups, Azure backs up only changed blocks.

Azure also supports encrypted VM backups using Storage Service Encryption (SSE) or Azure Disk Encryption. With SSE, Azure Storage automatically encrypts data at rest before storing it. Azure decrypts data during restore. Azure Disk Encryption encrypts both OS and data disks using either BitLocker encryption keys (BEKs) or Azure Key Vault key encryption (KEKs). Azure securely backs up both BEKs and KEKs and encrypts them within a key vault.<sup>85</sup>

### MySQL backup considerations

MySQL supports both physical and logical backups.

#### *Physical backups*

For physical backups, users can make raw copies of the database directories and files, using tools such as xtrabackup, mysqlbackup or Linux filesystem commands (tar, rsync, cp, etc.). Physical backups can be done when MySQL server is shut down. If the server is running, it is necessary to lock the server so that the database contents don't change during the backup (though third-party tools such as Percona Xtrabackup mitigate this challenge). Physical backups are typically faster than logical backups because they involve only copying files without conversion. To restore, you can copy file copies at the file system level back to their original locations with the same filesystem commands (tar, rsync and cp).<sup>86</sup>

#### *Logical backups*

To perform logical backups, you query the MySQL server for information about the database structure and content. Logical backups do not include log or configuration files or other related files if they are not part of the databases. You can perform logical backups using mysqldump while the MySQL server is running. To restore logical backups, you can use mysql client to process SQL-format files, or use the 'LOAD DATA' statement to process delimited-text files.<sup>87</sup>

#### *Full and incremental backups*

MySQL supports both full and incremental backups. A full backup is a copy of all database contents at a given point in time. You can take full backups using either physical or logical backup methods as we describe above. An incremental backup consists of changes made to the database during a given period. To do incremental backups, you must enable the server's binary log, which records all the changes the server made to the database.<sup>88</sup>



### Full and incremental (a.k.a. point-in-time) recovery

A full recovery restores the database instance to the state when the backup occurred. An incremental recovery restores all the changes after the full backup took place. MySQL supports point-in-time recovery by replaying the binary logs since a full backup.<sup>89</sup>

### Encrypting backups

You should ensure that your backup method of choice secures the data contained within it by way of encryption to protect against unauthorized access.

MySQL Enterprise Backup (mysqlbackup) supports encryption for backups. It uses a symmetric encryption algorithm, Advanced Encryption Standard (AES) 256, to encrypt user data with a key string provided by the user. You can manually create the key string or randomly generate it using tools such as OpenSSL.<sup>90</sup>

For example, to generate a random string using openssl:

```
openssl rand -hex 32
```

To encrypt a backup using mysqlbackup and the random key:

```
mysqlbackup --backup-file=/path/to/backup-file \  
--encrypt=AES-256 \  
--encrypt-key=your_encryption_key \  
restore
```

### Restoring backups

To restore an encrypted backup, you must provide the same key string to mysqlbackup:

```
mysqlbackup --backup-file=/path/to/backup-file \  
--decrypt=AES-256 \  
--decrypt-key=your_encryption_key \  
restore
```

## Monitoring and maintenance

Monitoring any application is a key component of your organization's planning and strategy. In the context of deploying MySQL to Microsoft Azure, you can use a variety of monitoring tools and techniques across cloud, OS, and database layers to optimize your configuration.

### Azure Monitor

Azure Monitor is a cloud monitoring solution offered by Microsoft that collects telemetry data from multiple types of resources and gives the user the ability to analyze, extract, and use the data in various ways, including alerting.

According to Microsoft, "Azure Monitor can monitor these types of resources in Azure, other clouds, or on-premises:

- Applications
- Virtual machines
- Guest operating systems
- Containers including Prometheus metrics
- Databases
- Security events in combination with Azure Sentinel
- Networking events and health in combination with Network Watcher
- Custom sources that use the APIs to get data into Azure Monitor"<sup>91</sup>

Azure Monitor uses a variety of techniques to collect data, including logs and metrics collection, agent-based collection, and third-party sources.

Azure Virtual Machines offers telemetry for CPU utilization, logical disk IOPS, and available memory when you enable Azure Virtual Machine Insights. For additional metrics, you can install the Microsoft Azure Monitor agent. For more details on agent installation, see <https://learn.microsoft.com/azure/azure-monitor/agents/azure-monitor-agent-manage?tabs=azure-cli#install>.

### Monitoring at the MySQL level

MySQL provides several built-in monitoring tools that allow you to monitor performance, resource utilization, query execution, and server health. Here are some of the built-in monitoring tools in MySQL:

- PERFORMANCE\_SCHEMA provides detailed insights into server activity, resource usage, and performance metrics by collecting data from internal MySQL server components. Users can use Performance Schema to monitor and analyze MySQL performance metrics, identify bottlenecks, and optimize database performance.<sup>92</sup>
- INFORMATION\_SCHEMA is a database within MySQL. It provides metadata and information about MySQL objects such as server such as the name of a database, table, column, indexes, and privileges. You can query the Information Schema tables and retrieve information about database status, storage engines, table sizes, index usage, user privileges, and more. It is a very useful tool to report database schema details, analyze database objects, and retrieve server configuration settings.<sup>93</sup>
- If you're using MySQL Enterprise Edition, MySQL Enterprise Monitor (MEM) offers additional real-time monitoring of MySQL databases, providing visibility into key performance metrics, resource utilization, query performance, and server status. For more information about MEM, please visit <https://www.mysql.com/products/enterprise/monitor.html>.

In addition to MySQL built-in tools and plugins, you can leverage Linux utilities such as top, mpstat, iostat, nmon, etc., to monitor various aspects like CPU usage, memory utilization, disk I/O, and network traffic on your MySQL server VMs.

## Other tools to consider

### Percona MySQL

Percona MySQL is a distribution of the MySQL database.<sup>94</sup> It is based on the open-source MySQL database and includes additional features, optimizations, and tools to enhance performance, scalability, and manageability for MySQL deployments. Percona MySQL includes tools such as the following:

- Percona XtraDB Cluster is an open-source high-availability solution for MySQL. It combines Percona Server for MySQL and Percona XtraBackup with the Galera library to enable synchronous multi-source replication. In an XtraDB cluster, each node contains the same set of data, which Percona XtraDB Cluster synchronizes across all nodes. Percona XtraDB Cluster executed queries locally on each node because all data is available locally without the need for remote access. This allows the cluster to continue functioning without data loss even if a node goes down. It is an excellent solution for read-intensive workloads, because it can distribute read queries across multiple nodes.<sup>95</sup>
- Percona XtraBackup is an open-source tool that enables hot backups for MySQL databases without locking the database during the process. It supports incremental backups, which capture only the changes since the last backup. It also supports point-in-time recovery, allowing you to restore the database to a specific point in time.<sup>96</sup> XtraBackup offers compression and encryption of backup files, enhancing security and reducing storage space requirements.<sup>97</sup> It supports various compression methods, including gzip, lz4, and bzip2. For encryption, it uses AES encryption to secure your backup files.<sup>98</sup>
- You can use the Percona Operator to automatically deploy your MySQL servers as containers in Azure Kubernetes Service (AKS) environment. The containers deployed by the Operator include Percona Server for MySQL, Percona XtraBackup, HAProxy, Orchestrator, and MySQL Router.<sup>99</sup> By using the Percona Operator for MySQL, you can leverage the power of AKS to efficiently manage and scale your MySQL databases, ensuring high availability and reliability.<sup>100</sup>

For a complete list of features and tools Percona offers, please visit <https://docs.percona.com/percona-for-mysql>.

### Charmed MySQL

Charmed MySQL is an enterprise-grade solution to secure and automate the deployment, maintenance, and upgrades of your MySQL databases. You can deploy it on Azure VMs and AKS.

Charmed MySQL automation is based on a curated list of open-source components including:

- MySQL Group Replication to provide high availability.<sup>101</sup>
- MySQL Router to provide automatic and transparent failovers for the connected applications.<sup>102</sup>
- A full monitoring and alerting solution based on Prometheus, Loki, and Grafana.<sup>103</sup>

Canonical also offers managed MySQL services and consulting for tailored solutions, including deployment, optimization, support, and training. For more information, please visit Canonical's MySQL page.<sup>104</sup>

## Conclusion

Choosing the right deployment model for MySQL on Azure depends on your organization's specific needs for control, flexibility, and management. Azure Database for MySQL – Flexible Server offers a fully managed PaaS platform that can simplify operations. The PaaS solutions could be ideal for applications requiring rapid deployment and scaling, minimal administrative overhead, and seamless scalability. Conversely, deploying MySQL on Azure IaaS provides greater control over the infrastructure. This approach offers flexibility to configure the operating system, storage, and networking according to customer needs and could be ideal for legacy applications, complex regulatory environments, or scenarios requiring higher customization. This guide presented how to use both Azure approaches as well as several recommendations on AMD EPYC™ CPU-based infrastructure to consider. Whether you want the simplicity of PaaS or maximum control with IaaS, Azure provides the tools and means to deploy MySQL databases effectively in the cloud.

- 
1. Wikipedia, "MySQL," accessed July 16, 2024, <https://en.wikipedia.org/wiki/MySQL>.
  2. Azure, "Supported versions for Azure Database for MySQL – Flexible Server," accessed September 9, 2024, <https://learn.microsoft.com/azure/mysql/flexible-server/concepts-supported-versions>.
  3. Azure, "Scheduled maintenance in Azure Database for MySQL – Flexible Server," accessed September 9, 2024, <https://learn.microsoft.com/azure/mysql/flexible-server/concepts-maintenance>.
  4. Azure, "What is Azure Database for MySQL – Flexible Server?" accessed September 19, 2024, <https://learn.microsoft.com/azure/mysql/flexible-server/overview#azure-regions>.
  5. Microsoft, "How GeekWire is serving the latest tech news to 2 million readers worldwide," accessed September 19, 2024, <https://customers.microsoft.com/story/geekwire>.
  6. Azure, "What's happening to Azure Database for MySQL – Single Server?" accessed September 5, 2024, <https://learn.microsoft.com/azure/mysql/migrate/whats-happening-to-mysql-single-server>.
  7. Azure, "What's happening to Azure Database for MySQL – Single Server?"
  8. Azure, "Azure Database for MySQL – Flexible Server service tiers," accessed August 30, 2024, <https://learn.microsoft.com/azure/mysql/flexible-server/concepts-service-tiers-storage>.
  9. Azure, "Azure Database for MySQL – Flexible Server service tiers."
  10. Azure, "B-series burstable virtual machine sizes," accessed September 6, 2024, <https://learn.microsoft.com/azure/virtual-machines/sizes-b-series-burstable>.
  11. Azure, "Virtual Machine series," accessed September 6, 2024, <https://azure.microsoft.com/pricing/details/virtual-machines/series/>.
  12. Sumit Jadhav, "Virtual machines in Azure," accessed September 9, 2024, <https://medium.com/@sumitjadhav6067/virtual-machines-in-azure-d51408147f4f>.
  13. Azure, "Virtual Machine series."
  14. Azure, "Accelerated logs feature in Azure Database for MySQL – Flexible Server," accessed September 19, 2024, <https://learn.microsoft.com/azure/mysql/flexible-server/concepts-accelerated-logs>.

- 
15. Azure, "Azure Database for MySQL – Flexible Server service tiers."
  16. Azure, "Dasv5 and Dadsv5-series" accessed August 30, 2024, <https://learn.microsoft.com/azure/virtual-machines/dasv5-dadsv5-series>.
  17. Azure, "Dadsv6 sizes series," accessed September 19, 2024, <https://learn.microsoft.com/azure/virtual-machines/sizes/general-purpose/dadsv6-series?tabs=sizebasic>.
  18. Azure, "Easv5 and Eadsv5-series," accessed August 30, 2024, <https://learn.microsoft.com/azure/virtual-machines/easv5-eadsv5-series#eadsv5-series>.
  19. Azure, "Azure Database for MySQL – Flexible Server service tiers."
  20. Azure, "Backup and restore in Azure Database for MySQL – Flexible Server," accessed September 9, 2024, <https://learn.microsoft.com/azure/mysql/flexible-server/concepts-backup-restore>.
  21. Azure, "Service Level Agreements (SLA) for Online Services," accessed October 4, 2024, <https://www.microsoft.com/licensing/docs/view/Service-Level-Agreements-SLA-for-Online-Services>.
  22. Azure, "High availability concepts in Azure Database for MySQL – Flexible Server," accessed October 4, 2024, <https://learn.microsoft.com/azure/mysql/flexible-server/concepts-high-availability>.
  23. Azure, "High availability concepts in Azure Database for MySQL – Flexible Server."
  24. Azure, "Service Level Agreements (SLA) for Online Services."
  25. Azure, "Data encryption with customer managed keys for Azure Database for MySQL – Flexible Server," accessed September 9, 2024, <https://learn.microsoft.com/azure/mysql/flexible-server/concepts-customer-managed-key>.
  26. Azure, "Scheduled maintenance in Azure Database for MySQL – Flexible Server."
  27. Azure, "Microsoft Entra authentication for Azure Database for MySQL – Flexible Server," accessed September 9, 2024, <https://learn.microsoft.com/azure/mysql/flexible-server/concepts-azure-ad-authentication>.
  28. Azure, "Private Network Access using virtual network integration for Azure Database for MySQL – Flexible Server," accessed September 9, 2024, <https://learn.microsoft.com/azure/mysql/flexible-server/concepts-networking-vnet>.
  29. Azure, "Backup and restore in Azure Database for MySQL – Flexible Server," accessed September 9, 2024, <https://learn.microsoft.com/azure/mysql/flexible-server/concepts-backup-restore>.
  30. Azure, "Monitor Azure Database for MySQL – Flexible Server," accessed September 9, 2024, <https://learn.microsoft.com/azure/mysql/flexible-server/concepts-monitoring>.
  31. Azure, "Error logs in Azure Database for MySQL – Flexible Server (Preview)," accessed September 5, 2024, <https://learn.microsoft.com/azure/mysql/flexible-server/concepts-error-logs>.
  32. Azure, "Compare support plans," accessed September 5, 2024, <https://azure.microsoft.com/support/plans/>.
  33. Azure, "Azure Support: Standard," accessed September 9, 2024, <https://azure.microsoft.com/support/plans/standard/>.
  34. Azure, "Azure Support: Professional Direct (ProDirect)," accessed September 9, 2024, <https://azure.microsoft.com/support/plans/prodirect/>.
  35. Microsoft, "Easv6 and Eadsv6-series (Preview)," accessed May 15, 2024, <https://learn.microsoft.com/azure/virtual-machines/easv6-eadsv6-series>.
  36. Microsoft, "Falsv6, Fasv6, and Famsv6-series (Preview)," accessed May 15, 2024, <https://learn.microsoft.com/azure/virtual-machines/fasv6-falsv6-series>.
  37. MySQL, "Supported Platforms: MySQL Database," accessed May 23, 2024, <https://www.mysql.com/support/supportedplatforms/database.html>.
  38. Microsoft, "Introduction to Azure managed disks," accessed May 15, 2024, <https://learn.microsoft.com/azure/virtual-machines/managed-disks-overview>.
  39. Microsoft, "Introduction to Azure managed disks," accessed May 17, 2024, <https://learn.microsoft.com/azure/virtual-machines/managed-disks-overview>.
  40. Microsoft, "Overview of Azure Disk Backup," accessed May 15, 2024, <https://learn.microsoft.com/azure/backup/disk-backup-overview>.
  41. Microsoft, "Create an incremental snapshot for managed disks," accessed June 13, 2024, <https://learn.microsoft.com/azure/virtual-machines/disks-incremental-snapshots>.
  42. Microsoft, "What is Azure Elastic SAN?" accessed May 15, 2024, <https://learn.microsoft.com/azure/storage/elastic-san/elastic-san-introduction>.

- 
43. Microsoft, "What is Azure Elastic SAN?" accessed May 15, 2024, <https://learn.microsoft.com/azure/storage/elastic-san/elastic-san-introduction>.
  44. Microsoft, "Azure Elastic SAN," accessed June 13, 2024, <https://azure.microsoft.com/products/storage/elastic-san>.
  45. MySQL, "8.10.3 The MySQL Query Cache," accessed May 23, 2024, <https://dev.mysql.com/doc/refman/5.7/en/query-cache.html>.
  46. MySQL, "10.10 Buffering and Caching," accessed May 23, 2024, <https://dev.mysql.com/doc/refman/8.0/en/buffering-caching.html>.
  47. Microsoft, "Azure premium storage: Design for high performance," accessed June 13, 2024, <https://learn.microsoft.com/azure/virtual-machines/premium-storage-performance>.
  48. Microsoft, "Configure Software RAID on Linux," accessed June 13, 2024, <https://learn.microsoft.com/previous-versions/azure/virtual-machines/linux/configure-raid>.
  49. MySQL, "18.11 Overview of MySQL Storage Engine Architecture," accessed May 24, 2024, <https://dev.mysql.com/doc/refman/8.0/en/pluggable-storage-overview.html>.
  50. Microsoft, "Sizes for virtual machines in Azure," accessed May 15, 2024, <https://learn.microsoft.com/azure/virtual-machines/sizes>.
  51. Microsoft, "Scale up an app in Azure App Service," accessed June 13, 2024, <https://learn.microsoft.com/azure/app-service/manage-scale-up>.
  52. Microsoft, "What are Virtual Machine Scale Sets?" accessed June 13, 2024, <https://learn.microsoft.com/azure/virtual-machine-scale-sets/overview>.
  53. Microsoft, "Availability sets overview," accessed June 13, 2024, <https://learn.microsoft.com/azure/virtual-machines/availability-set-overview>.
  54. Microsoft, "Best practices for achieving high availability with Azure virtual machines and managed disks," accessed July 17, 2024, <https://learn.microsoft.com/azure/virtual-machines/disks-high-availability>.
  55. Microsoft, "Best practices for achieving high availability with Azure virtual machines and managed disks,"
  56. Microsoft, "What are availability zones?" accessed June 13, 2024, <https://learn.microsoft.com/azure/reliability/availability-zones-overview>.
  57. "Azure Virtual Machines baseline architecture," accessed August 7, 2024, <https://learn.microsoft.com/azure/architecture/virtual-machines/baseline>.
  58. Microsoft, "What are Virtual Machine Scale Sets?" accessed June 13, 2024, <https://learn.microsoft.com/azure/virtual-machine-scale-sets/overview>.
  59. Microsoft, "Availability sets overview," accessed June 13, 2024, <https://learn.microsoft.com/azure/virtual-machines/availability-set-overview>.
  60. Microsoft, "Orchestration modes for Virtual Machine Scale Sets in Azure," accessed July 22, 2024, <https://learn.microsoft.com/azure/virtual-machine-scale-sets/virtual-machine-scale-sets-orchestration-modes#scale-sets-with-flexible-orchestration>.
  61. MySQL, "MySQL NBD Cluster: High Availability," accessed May 23, 2024, <https://www.mysql.com/products/cluster/availability.html>.
  62. MySQL, "Chapter 23 InnoDB Cluster," accessed May 23, 2024, <https://dev.mysql.com/doc/refman/8.0/en/mysql-innodb-cluster-introduction.html>.
  63. MySQL, "19.4.11 Delayed Replication," accessed May 23, 2024, <https://dev.mysql.com/doc/refman/8.0/en/replication-delayed.html>.
  64. MySQL, "19.2.1 Replication Formats," accessed May 23, 2024, <https://dev.mysql.com/doc/refman/8.3/en/replication-formats.html>.
  65. Microsoft, "Azure confidential virtual machines FAQ," accessed July 17, 2024, <https://learn.microsoft.com/azure/confidential-computing/confidential-vm-faq>.
  66. Microsoft, "ECasv5 and ECadsv5-series," accessed July 17, 2024, <https://learn.microsoft.com/azure/virtual-machines/ecasv5-ecadsv5-series>.
  67. Microsoft, "DCasv5 and DCadsv5-series confidential VMs," accessed July 17, 2024, <https://learn.microsoft.com/azure/virtual-machines/dcasv5-dcadsv5-series>.

- 
68. Microsoft, "About Azure Key Vault," accessed May 15, 2024, <https://learn.microsoft.com/azure/key-vault/general/overview>.
  69. Microsoft, "Azure Disk Encryption for Windows VMs," accessed May 15, 2024, <https://learn.microsoft.com/azure/virtual-machines/windows/disk-encryption-overview>.
  70. Microsoft, "Azure Disk Encryption for Linux VMs," accessed May 15, 2024, <https://learn.microsoft.com/azure/virtual-machines/linux/disk-encryption-overview>.
  71. Microsoft, "Azure Virtual Network," accessed May 15, 2024, <https://azure.microsoft.com/products/virtual-network/>.
  72. Cloudflare, "What is a DDoS attack?" accessed May 15, 2024, <https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/>.
  73. Microsoft, "Azure DDoS Protection," accessed May 15, 2024, <https://azure.microsoft.com/products/ddos-protection>.
  74. Microsoft, "Azure Firewall," accessed May 15, 2024, <https://azure.microsoft.com/products/azure-firewall/>.
  75. Microsoft, "What is Azure Load Balancer?" accessed May 15, 2024, <https://learn.microsoft.com/azure/load-balancer/load-balancer-overview>.
  76. MySQL, "4.2 Privileges Provided by MySQL," accessed May 23, 2024, <https://dev.mysql.com/doc/mysql-security-excerpt/8.0/en/privileges-provided.html>.
  77. MySQL, "Chapter 11 Enabling Authentication," accessed May 23, 2024, <https://dev.mysql.com/doc/mysql-secure-deployment-guide/5.7/en/secure-deployment-configure-authentication.html>.
  78. MySQL, "8.4.1.7 LDAP Pluggable Authentication," accessed May 23, 2024, <https://dev.mysql.com/doc/refman/8.0/en/ldap-pluggable-authentication.html>.
  79. MySQL, "8.4.1.5 PAM Pluggable Authentic," accessed May 23, 2024, <https://dev.mysql.com/doc/refman/8.0/en/pam-pluggable-authentication.html>.
  80. MySQL, "8.4.1.8 Kerberos Pluggable Authentic," accessed May 23, 2024, <https://dev.mysql.com/doc/refman/8.0/en/kerberos-pluggable-authentication.html>.
  81. MySQL, "8.3.1 Configuring MySQL to Use Encrypted Connections," accessed May 23, 2024, <https://dev.mysql.com/doc/refman/8.3/en/using-encrypted-connections.html>.
  82. MySQL, "17.3 InnoDB Data-at-Rest Encryption," accessed May 23, 2024, <https://dev.mysql.com/doc/refman/8.0/en/innodb-data-encryption.html>.
  83. MySQL, "5.3.1 Standard TCP/IP Connection Method," accessed May 23, 2024, <https://dev.mysql.com/doc/workbench/en/wb-mysql-connections-methods-standard.html>.
  84. Microsoft, "An overview of Azure VM backup," accessed May 15, 2024, <https://learn.microsoft.com/azure/backup/backup-azure-vms-introduction>.
  85. Microsoft, "An overview of Azure VM backup."
  86. MySQL, "9.1. Backup and Recovery Types," accessed May 23, 2024, <https://dev.mysql.com/doc/refman/8.0/en/backup-types.html>.
  87. MySQL, "9.1. Backup and Recovery Types."
  88. MySQL, "9.2 Database Backup Methods," accessed May 23, 2024, <https://dev.mysql.com/doc/refman/8.0/en/backup-methods.html>.
  89. MySQL, "9.5 Point-in-Time (Incremental) Recovery," accessed May 23, 2024, <https://dev.mysql.com/doc/refman/8.0/en/point-in-time-recovery.html>.
  90. MySQL, "Chapter 10 Encryption for Backups," accessed May 23, 2024, <https://dev.mysql.com/doc/mysql-enterprise-backup/8.3/en/meb-encryption.html>.
  91. Microsoft, "Azure Monitor overview," accessed May 15, 2024, <https://learn.microsoft.com/azure/azure-monitor/overview>.
  92. MySQL, "Chapter 29 MySQL Performance Schema," accessed May 23, 2024, <https://dev.mysql.com/doc/refman/8.3/en/pCperformance-schema.html>.
  93. MySQL, "Chapter 2 Introduction," accessed May 23, 2024, <https://dev.mysql.com/doc/mysql-infoschema-excerpt/8.3/en/information-schema-introduction.html>.

- 
94. Percona, "Percona Software for MySQL," accessed May 23, 2024, <https://www.percona.com/mysql/software>.
  95. Percona, "Percona XtraDB Cluster 8.0 Documentation," accessed May 23, 2024, <https://docs.percona.com/percona-xtradb-cluster/8.0/>.
  96. Percona, "Percona XtraBackup 8.3 Documentation," accessed May 23, 2024, <https://docs.percona.com/percona-xtrabackup/innovation-release/>.
  97. Percona, "Percona XtraBackup 8.3 Documentation," accessed May 23, 2024, <https://docs.percona.com/percona-xtrabackup/innovation-release/>.
  98. Percona, "Percona XtraBackup 8.3 Documentation," accessed May 23, 2024, <https://docs.percona.com/percona-xtrabackup/innovation-release/>.
  99. Percona, "Percona Operator for MySQL based on Percona Server for MySQL," accessed May 23, 2024, <https://docs.percona.com/percona-operator-for-mysql/ps/>.
  100. Percona, "Percona Operator for MySQL based on Percona Server for MySQL," accessed May 23, 2024, <https://docs.percona.com/percona-operator-for-mysql/ps/>.
  101. Canonical, "Scale your Charmed MySQL," accessed July 17, 2024, <https://charmhub.io/mysql/docs/t-managing-units>.
  102. Canonical, "MySQL Router," accessed July 17, 2024, <https://charmhub.io/mysql-router>.
  103. Canonical, "Enable monitoring," accessed July 17, 2024, <https://canonical.com/data/docs/mysql/iaas/h-enable-monitoring>.
  104. Canonical, "MySQL operations, simplified," accessed July 17, 2024, <https://canonical.com/data/mysql>.



## Appendix

### Sample ARM Template for deploying Azure Database for MySQL

#### template.json

```
{
  "$schema": "http://schema.management.azure.com/schemas/2014-04-01-preview/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "apiVersion": {
      "type": "string",
      "defaultValue": "2021-05-01"
    },
    "administratorLogin": {
      "type": "string"
    },
    "administratorLoginPassword": {
      "type": "securestring"
    },
    "location": {
      "type": "string"
    },
    "serverName": {
      "type": "string"
    },
    "serverEdition": {
      "type": "string"
    },
    "vCores": {
      "type": "int",
      "defaultValue": 4
    },
    "storageSizeGB": {
      "type": "int"
    },
    "haEnabled": {
      "type": "string",
      "defaultValue": "Disabled"
    },
    "availabilityZone": {
      "type": "string",
      "defaultValue": ""
    },
    "standbyAvailabilityZone": {
      "type": "string"
    },
    "version": {
      "type": "string"
    },
    "tags": {
      "type": "object",
      "defaultValue": {}
    },
    "firewallRules": {
      "type": "object",
      "defaultValue": {}
    },
    "backupRetentionDays": {
      "type": "int"
    },
    "geoRedundantBackup": {
      "type": "string"
    },
    "vmName": {
      "type": "string",
      "defaultValue": "Standard_B1ms"
    }
  },
```

```

    "storageIops": {
      "type": "int"
    },
    "storageAutogrow": {
      "type": "string",
      "defaultValue": "Enabled"
    },
    "autoIoScaling": {
      "type": "string",
      "defaultValue": "Disabled"
    },
    "identityData": {
      "type": "object",
      "defaultValue": {}
    },
    "dataEncryptionData": {
      "type": "object",
      "defaultValue": {}
    },
    "serverParameters": {
      "type": "array",
      "defaultValue": []
    },
    "aadEnabled": {
      "type": "bool",
      "defaultValue": false
    },
    "aadData": {
      "type": "object",
      "defaultValue": {}
    },
    "guid": {
      "type": "string",
      "defaultValue": "[newGuid()]"
    },
    "network": {
      "type": "object",
      "defaultValue": {}
    },
    "firewallRulesAPIVersion": {
      "type": "string",
      "defaultValue": "2022-01-01"
    },
    "acceleratedLogs": {
      "type": "string",
      "defaultValue": "Disabled"
    }
  },
  "variables": {
    "api": "[parameters('apiVersion')]",
    "firewallRules": "[parameters('firewallRules').rules]",
    "serverParameters": "[parameters('serverParameters')]"
  },
  "resources": [
    {
      "apiVersion": "[variables('api')]",
      "location": "[parameters('location')]",
      "name": "[parameters('serverName')]",
      "identity": "[if(empty(parameters('identityData')), json('null'), parameters('identityData'))]",
      "properties": {
        "createMode": "Default",
        "version": "[parameters('version')]",
        "administratorLogin": "[parameters('administratorLogin')]",
        "administratorLoginPassword": "[parameters('administratorLoginPassword')]",
        "Network": "[if(empty(parameters('network')), json('null'), parameters('network'))]",
        "Storage": {
          "StorageSizeGB": "[parameters('storageSizeGB')]",
          "Iops": "[parameters('storageIops')]",
          "Autogrow": "[parameters('storageAutogrow')]",
          "AutoIoScaling": "[parameters('autoIoScaling')]",
          "LogOnDisk": "[parameters('acceleratedLogs')]"
        }
      }
    }
  ]
}

```

```

    },
    "Backup": {
      "backupRetentionDays": "[parameters('backupRetentionDays')]",
      "geoRedundantBackup": "[parameters('geoRedundantBackup')]"
    },
    "availabilityZone": "[parameters('availabilityZone')]",
    "highAvailability": {
      "mode": "[parameters('haEnabled')]",
      "standbyAvailabilityZone": "[parameters('standbyAvailabilityZone')]"
    },
    "dataEncryption": "[if(empty(parameters('dataEncryptionData')), json('null'), parameters('dataEncryptionData'))]"
  },
  "sku": {
    "name": "[parameters('vmName')]",
    "tier": "[parameters('serverEdition')]",
    "capacity": "[parameters('vCores')]"
  },
  "tags": "[parameters('tags')]",
  "type": "Microsoft.DBforMySQL/flexibleServers"
},
{
  "condition": "[greater(length(variables('firewallRules')), 0)]",
  "type": "Microsoft.Resources/deployments",
  "apiVersion": "2019-08-01",
  "name": "[concat('firewallRules-', parameters('guid'), '-', copyIndex())]",
  "copy": {
    "count": "[if(greater(length(variables('firewallRules')), 0), length(variables('firewallRules')), 1)]",
    "mode": "Serial",
    "name": "firewallRulesIterator"
  },
  "dependsOn": [
    "[concat('Microsoft.DBforMySQL/flexibleServers/', parameters('serverName'))]",
    "[concat('Microsoft.Resources/deployments/addAdmins-', parameters('guid'))]"
  ],
  "properties": {
    "mode": "Incremental",
    "template": {
      "$schema": "http://schema.management.azure.com/schemas/2014-04-01-preview/deploymentTemplate.json#",
      "contentVersion": "1.0.0.0",
      "resources": [
        {
          "type": "Microsoft.DBforMySQL/flexibleServers/firewallRules",
          "name": "[concat(parameters('serverName'), '/', variables('firewallRules')[copyIndex()].name)]",
          "apiVersion": "[parameters('firewallRulesAPIVersion')]",
          "properties": {
            "startIPAddress": "[variables('firewallRules')[copyIndex()].startIPAddress]",
            "endIPAddress": "[variables('firewallRules')[copyIndex()].endIPAddress]"
          }
        }
      ]
    }
  }
},
{
  "condition": "[parameters('aadEnabled')]",
  "type": "Microsoft.Resources/deployments",
  "apiVersion": "2019-08-01",
  "name": "[concat('addAdmins-', parameters('guid'))]",
  "dependsOn": [
    "[concat('Microsoft.DBforMySQL/flexibleServers/', parameters('serverName'))]"
  ],
  "properties": {
    "mode": "Incremental",
    "template": {
      "$schema": "http://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
      "contentVersion": "1.0.0.0",

```

```

        "resources": [
            {
                "type": "Microsoft.DBforMySQL/flexibleServers/administrators",
                "name": "[concat(parameters('serverName'), '/ActiveDirectory')]",
                "apiVersion": "[variables('api')]",
                "properties": {
                    "administratorType": "[parameters('aadData').administratorType]",
                    "identityResourceId": "[parameters('aadData').identityResourceId]",
                    "login": "[parameters('aadData').login]",
                    "sid": "[parameters('aadData').sid]",
                    "tenantId": "[parameters('aadData').tenantId]"
                }
            }
        ]
    },
    {
        "condition": "[and(greater(length(variables('serverParameters')), 0),
parameters('aadEnabled'))]",
        "type": "Microsoft.Resources/deployments",
        "apiVersion": "2019-08-01",
        "copy": {
            "count": "[if(greater(length(variables('serverParameters')), 0), length(variables('serve
rParameters')), 1)]",
            "mode": "serial",
            "name": "serverParametersIterator"
        },
        "dependsOn": [
            "[concat('Microsoft.DBforMySQL/flexibleServers/', parameters('serverName'))]",
            "[concat('Microsoft.Resources/deployments/addAdmins-', parameters('guid'))]"
        ],
        "name": "[concat('serverParameters-', copyIndex(), '-', parameters('guid'))]",
        "properties": {
            "mode": "Incremental",
            "template": {
                "$schema": "http://schema.management.azure.com/schemas/2014-04-01-preview/
deploymentTemplate.json#",
                "contentVersion": "1.0.0.0",
                "resources": [
                    {
                        "type": "Microsoft.DBforMySQL/flexibleServers/configurations",
                        "name": "[concat(parameters('serverName'), '/', variables('serverParameters')
[copyIndex()].name)]",
                        "apiVersion": "[variables('api')]",
                        "properties": {
                            "value": "[variables('serverParameters')[copyIndex()].value]",
                            "source": "[variables('serverParameters')[copyIndex()].source]"
                        }
                    }
                ]
            }
        }
    }
]
}

```

## parameters.json

```
{
  "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "serverName": {
      "value": "pt-admysql-sample-deployment"
    },
    "location": {
      "value": "eastus"
    },
    "serverEdition": {
      "value": "GeneralPurpose"
    },
    "vCores": {
      "value": 16
    },
    "storageSizeGB": {
      "value": 64
    },
    "haEnabled": {
      "value": "ZoneRedundant"
    },
    "firewallRules": {
      "value": {
        "rules": [
          {
            "name": "AllowAllAzureServicesAndResourcesWithinAzureIps_2024-10-8_12-45-23",
            "endIPAddress": "0.0.0.0",
            "startIPAddress": "0.0.0.0"
          },
          {
            "name": "office-access",
            "startIPAddress": "48.96.71.70",
            "endIPAddress": "48.96.71.70"
          }
        ]
      }
    },
    "backupRetentionDays": {
      "value": 14
    },
    "geoRedundantBackup": {
      "value": "Enabled"
    },
    "availabilityZone": {
      "value": ""
    },
    "vmName": {
      "value": "Standard_D16ads_v5"
    },
    "storageIops": {
      "value": 0
    },
    "storageAutogrow": {
      "value": "Enabled"
    },
    "acceleratedLogs": {
      "value": "Disabled"
    },
    "standbyAvailabilityZone": {
      "value": ""
    },
    "autoIoScaling": {
      "value": "Enabled"
    },
    "guid": {
      "value": "1a0469e4b2af466e9825660da4b45b7c"
    },
    "serverParameters": {
      "value": []
    }
  }
}
```

```
    },
    "apiVersion": {
      "value": "2023-06-01-preview"
    },
    "network": {
      "value": {
        "publicNetworkAccess": "Enabled"
      }
    },
    "tags": {
      "value": {}
    },
    "administratorLogin": {
      "value": "ptadmin"
    },
    "administratorLoginPassword": {
      "value": null
    },
    "version": {
      "value": "8.0.21"
    }
  }
}
```

This project was commissioned by Microsoft and AMD.



Facts matter.®

Principled Technologies is a registered trademark of Principled Technologies, Inc. All other product names are the trademarks of their respective owners.

**DISCLAIMER OF WARRANTIES; LIMITATION OF LIABILITY:**

Principled Technologies, Inc. has made reasonable efforts to ensure the accuracy and validity of its testing, however, Principled Technologies, Inc. specifically disclaims any warranty, expressed or implied, relating to the test results and analysis, their accuracy, completeness or quality, including any implied warranty of fitness for any particular purpose. All persons or entities relying on the results of any testing do so at their own risk, and agree that Principled Technologies, Inc., its employees and its subcontractors shall have no liability whatsoever from any claim of loss or damage on account of any alleged error or defect in any testing procedure or result.

In no event shall Principled Technologies, Inc. be liable for indirect, special, incidental, or consequential damages in connection with its testing, even if advised of the possibility of such damages. In no event shall Principled Technologies, Inc.'s liability, including for direct damages, exceed the amounts paid in connection with Principled Technologies, Inc.'s testing. Customer's sole and exclusive remedies are as set forth herein.