

Creating the Azure SQL Database

Azure SQL Database is a fully managed relational database service provided by Microsoft on the Azure cloud platform. It is based on SQL Server and offers features like high availability, automated backups, scaling, and security, without the need to manage the underlying infrastructure. It allows users to store and manage data in the cloud with minimal administrative effort, providing automatic updates and patching.

Azure SQL Database is designed for cloud applications that require high-performance, scalability, and reliability. It supports a variety of workloads, including web applications, data warehousing, and enterprise solutions. It offers several deployment options, such as single databases, elastic pools (for multiple databases), and managed instances (for near-complete SQL Server compatibility).

Use Cases of Azure SQL Database

1. **Web and Mobile Applications:** Azure SQL Database is commonly used for storing and managing data for web and mobile applications, offering scalable performance and reliable availability.
2. **Business Applications:** It is ideal for enterprise business applications like customer relationship management (CRM) and enterprise resource planning (ERP) systems that require a robust and scalable database solution.
3. **Data Warehousing and Analytics:** Azure SQL Database can be used to store and analyze large amounts of data for business intelligence, supporting data warehousing solutions and integration with Power BI and Azure Synapse Analytics.
4. **SaaS Applications:** It's widely used for Software-as-a-Service (SaaS) applications, where multi-tenant architectures require secure and scalable database solutions with automatic backups and high availability.
5. **Backup and Disaster Recovery:** Azure SQL Database supports disaster recovery scenarios, enabling automated backups and geo-redundancy for business continuity.

The end goal of this lab is to successfully deploy an Azure SQL Database on the Azure cloud platform. This involves selecting a resource group, providing a unique database name, and configuring a server with SQL authentication. We also choose the appropriate region, cost-effective pricing model, and backup options, while enabling a public endpoint for database access. Additionally, a firewall rule is set to allow access from the current client device. After reviewing the configuration, we proceed to create and deploy the SQL database, ensuring a fully functional, scalable, and secure database solution in the cloud.

To begin with the lab

1. In previous lab we deploy a virtual machine. In this lab, we are going to deploy a SQL database. Go to the SQL Databases section in the Azure portal and click on Create.
2. Select an existing resource group or create a new one.

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ MSDN Platforms Subscription ▼

Resource group * ⓘ Select a resource group ▼
[Create new](#)

Database details

3. Provide a name for the database.

Database details

Enter required settings for this database, including picking a logical server and configuring the compute and storage resources

Database name * appdb ✓

Server * ⓘ Select a server ▼
[Create new](#)

4. Choose a unique server name.
5. Select the Azure region (e.g., North Europe) to keep all resources in one location.

Create SQL Database Server

Microsoft

Server details



Enter required settings for this server, including providing a name and location. This server will be created in the same subscription and resource group as your database.



Server name * demoserver6999 ✓
.database.windows.net




Location * (Europe) North Europe ▼

Authentication

6. Choose SQL authentication and provide the server admin login credentials (username and password) for accessing the database server.

 Azure Active Directory (Azure AD) is now Microsoft Entra ID. [Learn more](#) 


Select your preferred authentication methods for accessing this server. Create a server admin login and password to access your server with SQL authentication, select only Microsoft Entra authentication [Learn more](#)  using an existing Microsoft Entra user, group, or application as Microsoft Entra admin [Learn more](#) , or select both SQL and Microsoft Entra authentication.

Authentication method	<input type="radio"/> Use Microsoft Entra-only authentication <input type="radio"/> Use both SQL and Microsoft Entra authentication <input checked="" type="radio"/> Use SQL authentication
Server admin login *	<input type="text" value="sqlserver"/> 
Password *	<input type="password" value="....."/> 
Confirm password *	<input type="password" value="....."/> 

OK

7. Basic Model is selected for cost efficiency, as it offers the least expensive option.

Want to use SQL elastic pool? ⓘ	<input type="radio"/> Yes <input checked="" type="radio"/> No
Workload environment	<input checked="" type="radio"/> Development <input type="radio"/> Production

 Default settings provided for Development workloads. Configurations can be modified as needed.

Compute + storage * ⓘ	<div>Basic 2 GB storage Configure database</div>
-----------------------	---

8. Select Locally-redundant backup storage to save on backup storage costs.

Backup storage redundancy

Choose how your PITR and LTR backups are replicated. Geo restore or ability to recover from regional outage is only available when geo-redundant storage is selected.

Backup storage redundancy ⓘ	<input checked="" type="radio"/> Locally-redundant backup storage <input type="radio"/> Zone-redundant backup storage <input type="radio"/> Geo-redundant backup storage <input type="radio"/> Geo-Zone-redundant backup storage
-----------------------------	---

Review + create

Next : Networking >

9. Choose a public endpoint to allow access to the database.

10. Set a firewall rule to allow the **current client IP address** (the device used to connect to the database).

Basics **Networking** Security Additional settings Tags Review + create

Configure network access and connectivity for your server. The configuration selected below will apply to the selected server 'sqlserver6999' and all databases it manages. [Learn more](#)

Network connectivity

Choose an option for configuring connectivity to your server via public endpoint or private endpoint. Choosing no access creates with defaults and you can configure connection method after server creation. [Learn more](#)

Connectivity method * ⓘ
☐ No access
☒ Public endpoint
☐ Private endpoint

Firewall rules

Setting 'Allow Azure services and resources to access this server' to Yes allows communications from all resources inside the Azure boundary, that may or may not be part of your subscription. [Learn more](#)
Setting 'Add current client IP address' to Yes will add an entry for your client IP address to the server firewall.

Allow Azure services and resources to access this server *
☐ No ☒ Yes

Add current client IP address *
☐ No ☒ Yes

Connection policy

Configure how clients communicate with your SQL database server. [Learn more](#)

Connection policy ⓘ
☒ Default - Uses Redirect policy for all client connections originating inside of Azure (except Private Endpoint connections) and Proxy for all client connections originating outside Azure
☐ Proxy - All connections are proxied via the Azure SQL Database gateways

Review + create

< Previous

Next : Security >

11. After configuring security and additional settings, click on **Review + Create**.
12. Review the configuration, then click **Create** to begin the deployment.

✓ Your deployment is complete



Deployment name : Microsoft.SQLDatabase.newDatabase...

Start time : 12/17/2024, 1:30:06 PM

Subscription : [MSDN Platforms Subscription](#)

Correlation ID : 57ec337d-e4bc-4ea3-bb56-95e819b57...

Resource group : [Demorg](#)

> Deployment details

✓ Next steps

[Go to resource](#)