

Migrate SQL Server workloads to Azure SQL Managed Instance

1. Introduction

<https://learn.microsoft.com/en-us/training/modules/migrate-sql-workloads-azure-managed-instances/1-introduction>

Introduction

Completed

Azure SQL managed instances make it easy and quick to lift and shift a database from on-premises into the cloud.

Suppose you work for a sports clothing retail company. Your organization has been moving many of its systems into Azure to realize improved cost-of-ownership and better availability. So far, you haven't considered migrating your business-critical databases to Azure because you've heard that databases often need modification to work in Azure SQL.

You want to evaluate Azure's database hosting solutions, choose the best one to host your database, and do the migration. You'd prefer to migrate with as little database administrator and developer time as possible to keep the project budget low.

Easy migration: nearly 100% like SQL Server

Data migration

- Native backup/restore
- Configurable DB file layout
- DMS (migrations at scale)

Security

- Integrated Auth (Azure AD)
- Encryption (TDE, AE)
- SQL Audit
- Row-Level Security
- Dynamic Data Masking

Programmability

- Global temp tables
- Cross-database queries and transactions
- Linked servers
- CLR modules

Operational

- DMVs & XEvents
- Query Store
- SQL Agent
- DB Mail (external SMTP)

Scenario enablers

- Service Broker
- Change Data Capture
- Transactional Replication

In this module, you'll learn about Azure SQL Managed Instance, how it differs from other database solutions available in Azure, and how to choose the best cloud database solution for your project.

2. Evaluate migration scenarios

<https://learn.microsoft.com/en-us/training/modules/migrate-sql-workloads-azure-managed-instances/2-evaluate-migration-scenarios>

Evaluate migration scenarios

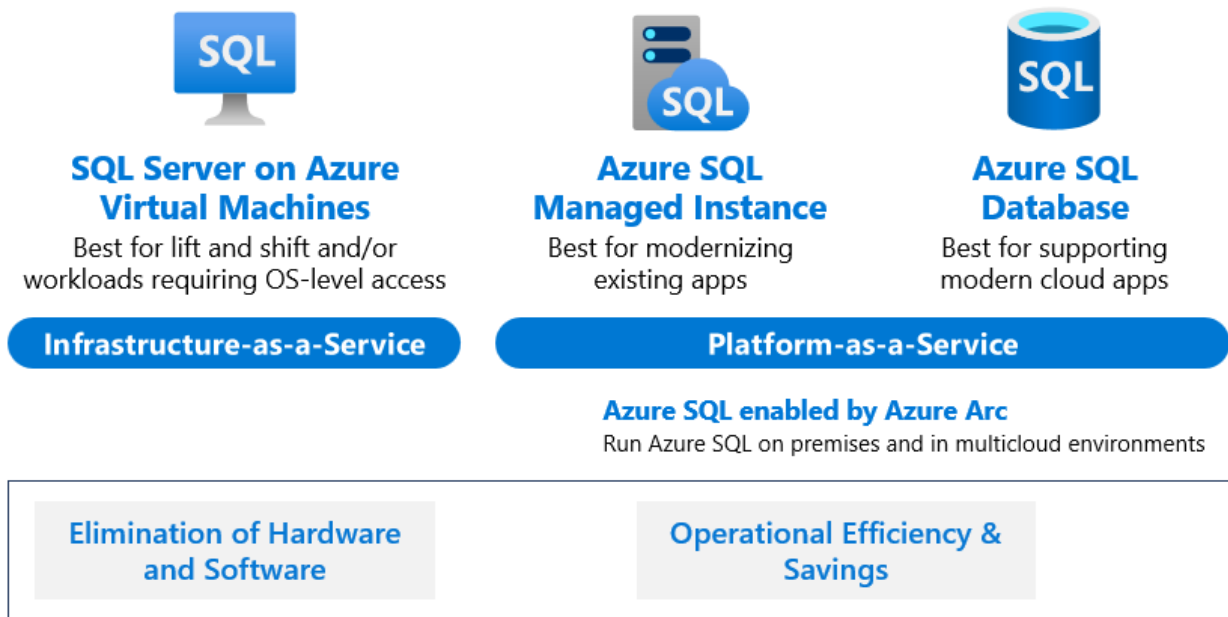
Completed

Azure SQL managed instance is designed to make it easy to host existing databases in the cloud by providing almost 100 percent compatibility with on-premises versions of SQL Server.

In your sports clothing company, you have a database that stores the product details for your entire catalog. The website uses the database to display product details to customers, by the sales representatives' smartphone apps to keep them informed about the catalog, and by a data analysis solution to populate product dimensions in a data cube. The database is considered business-critical by the board of directors. You've been asked to migrate this database to the cloud so the systems that depend on it need as little modification as possible. You want to evaluate Azure SQL Managed Instance for this project.

What is Azure SQL Managed Instance?

The Azure SQL platform as a service (PaaS) family includes the Azure SQL Database and Azure SQL Managed Instance. The goal of Azure SQL Managed Instance is to provide SQL Server applications with a fully managed PaaS experience in the Azure cloud.



Azure SQL Managed Instance is designed to enable a *lift and shift* solution for customers. The managed instance looks to bring applications, databases, and supporting technologies to Azure PaaS. Previously, without SQL Managed Instance, migration scenarios where an organization's application required access to any technology outside of the database (for example SQL Agent jobs, cross database joins, and SQL Server Integration Services) would be prevented from moving to the cloud. The only way a DBA or developer could migrate an on-premises application would be to employ one of the following approaches:

- Move the database and supporting technologies to an infrastructure as a service (IaaS) model.
- Rewrite the application with a fully PaaS model on Azure SQL Database, with extra development to address migration blockers.

The decision to migrate applications to Azure often hinges on whether an organization has the resources to adapt their application to Azure's PaaS model and manage the application code, as vendor support for modifications is typically limited. So, many opt for SQL Server on IaaS to use the full SQL Server experience without the need to overhaul existing applications. Despite Azure SQL Database's capabilities, the heavy dependence of many applications on technologies outside its scope presents challenges. However, SQL Managed Instance, code-named "*cloud lifter*," aims to overcome these hurdles, facilitating the migration to a SQL-based PaaS solution in Azure without needing application redesign.

Review key features

The most important features of SQL Managed Instance include:

Key Features	Description
Backwards compatibility	Managed instance provides backward compatibility to SQL Server 2008 databases. Direct migration from SQL Server 2005 database servers is also supported, with the compatibility level for migrated SQL Server 2005 databases being updated to SQL Server 2008.
Easy lift and shift	Managed instance has close to 100 percent compatibility to SQL Server. This compatibility includes core SQL Server components, programmability enhancements, instance-scoped features, such as cross database joins, and management tools that most existing SQL-based applications need to function correctly.
Fully managed PaaS	PaaS benefits include removing the need for managing hardware and all the overhead that comes from doing physical maintenance on SQL Server servers. You also have the benefits of quickly scaling up and scaling down, and provisioning resources in the cloud. SQL Managed Instance is built on the SQL Server engine, so it's always up to date with the latest SQL features and functionality.
Security features	You can enable security features at the SQL Managed Instance level just as you do at the database level. These features include the Vulnerability Assessment and the Advance Threat Protection settings. Finally, at the managed instance level, you can configure Transparent Data Encryption (TDE) and whether you want to bring your own key (BYOK) for encryption.
Secure network isolation	One of the unique aspects of managed instance, network security isolation is where managed instance has complete security isolation from any other tenant in the Azure cloud. In a typical default deployment SQL endpoint, managed instance is solely exposed through a private IP address that only allows connectivity from private Azure networks or hybrid networks. For on-premises applications to connect to managed instance, you'd need an Azure ExpressRoute configuration or a VPN gateway.
Instance failover groups	An instance failover group is a set of databases managed by a single database server, or within a single managed instance, that can fail over as a unit to another region. You use instance failover groups when all or some of the primary databases have gone offline due to an outage in the primary region.

Migration options supported

There are two modes of migration to Azure SQL Managed Instance: **online** and **offline**. The online mode has minimal or no downtime, while the offline mode experiences downtime during the migration process.

- **Log Replay Service.** It's an online migration option, and used when you need more control of your database migration project.
- **Managed Instance link..** The Managed Instance link, using distributed availability groups, securely extends your data estate by replicating data almost instantly (online) between any hosted SQL Server and Azure SQL Managed Instance, and vice versa.

- **SQL Server migration in Azure Arc.** When your SQL Server instance is enabled by Azure Arc, you can migrate databases to SQL Managed Instance directly from the Azure portal using either the Managed Instance link or Log Replay Service methods.
- **Native backup and restore.** Backup and restore are a simple migration method favored by many SQL Server professionals. It's the easiest migration option for customers who can provide full database backups to Azure Storage.
- **Transactional replication.** Transactional replication is a way to move data between continuously connected database servers. It's best to be used for online or offline migration of large and complex databases.

While most of the tools facilitate [migration to Azure SQL Database](#) as well, there are some that are exclusively supported by SQL Managed Instance. In the next units, we'll learn about a few of them in more detail.

Tip

Learn more about how to [design a SQL Server migration strategy](#).

3. Migrate using Azure Arc

<https://learn.microsoft.com/en-us/training/modules/migrate-sql-workloads-azure-managed-instances/3-azure-arc>

Migrate using Azure Arc

Completed

Azure Arc provides a streamlined pathway to migrate SQL Server databases to Azure SQL Managed Instance directly from the Azure portal. When your SQL Server instance is enabled by Azure Arc, you gain access to integrated assessment, target selection, and migration capabilities—all managed through a unified experience.

In our bicycle manufacturing company scenario, the distribution team runs business-critical databases that require minimal downtime during migration. The team wants to use the Azure portal for a simplified migration experience while maintaining control over the migration method.

Understand how to migrate SQL Server with Azure Arc

When your SQL Server instance is enabled by Azure Arc, the Azure portal becomes your central hub for database migration to SQL Managed Instance. The migration process guides you through four key stages:

1. **Assess source instance** - Evaluate your SQL Server instance to determine migration readiness
2. **Select target** - Choose or create your SQL Managed Instance destination
3. **Migrate data** - Execute the migration using your chosen method
4. **Monitor and cutover** - Track progress and complete the migration

Discovery of SQL Server instances and generation of readiness reports happen automatically every weekend, but you can also trigger assessments manually at any time. The assessment evaluates feature compatibility, identifies potential migration blockers, and provides right-sized recommendations for your target SQL Managed Instance configuration.

Database migration through Azure Arc is available for SQL Server 2012 and later versions, with the specific migration method determining exact version requirements.

Benefits of Azure Arc for migration

Azure Arc simplifies the migration journey by consolidating the entire process within the Azure portal. Rather than switching between multiple tools and interfaces, you can assess readiness, select or create your target instance, execute the migration, and monitor progress from a single unified experience. The automated assessment capabilities evaluate feature compatibility and identify potential migration blockers, while also providing right-sized recommendations for your target SQL Managed Instance configuration based on your workload characteristics.

Microsoft Copilot is integrated into the Azure portal migration experience to assist you throughout the process. You can interactively chat with Copilot to get help comparing migration methods, understanding assessments, starting migrations, and monitoring progress—making it easier to choose and execute the right approach for your scenario.

Additionally, Azure Arc provides visibility into your overall migration status, showing the number of databases migrated, migrations in progress, and your recommended target configuration. This centralized view helps you track progress across your data estate and ensures nothing is overlooked during large-scale migration projects.

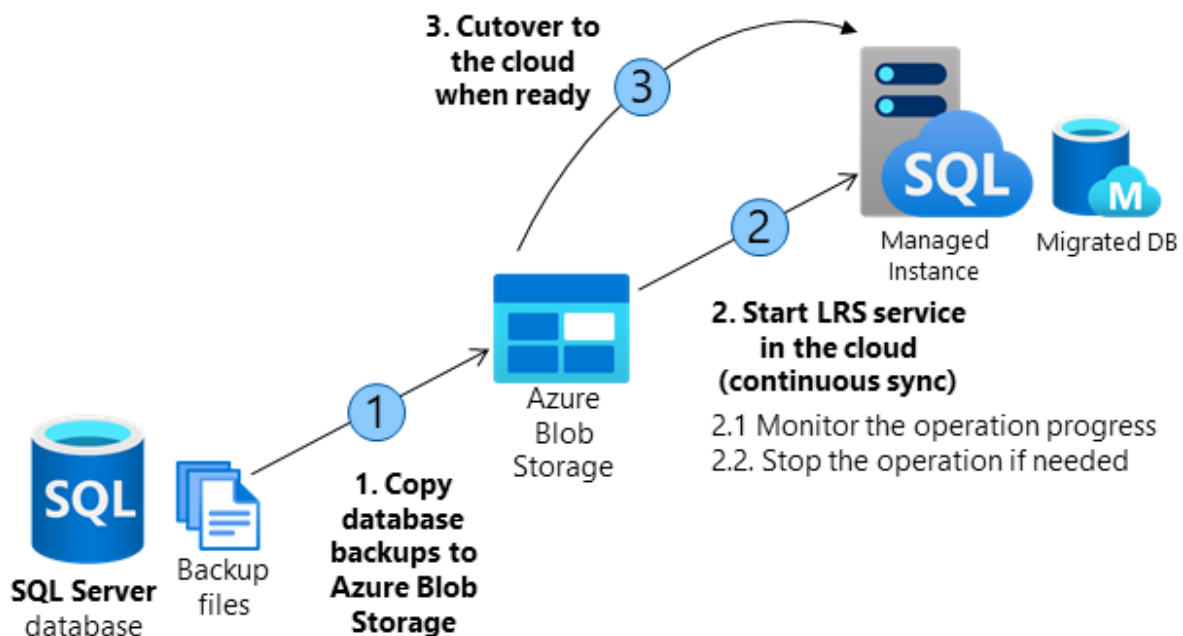
For detailed guidance on migrating SQL Server databases to Azure SQL Managed Instance using Azure Arc, see [Migration to Azure SQL Managed Instance](#).

4. Use Log Replay Service (LRS) to migrate

Use Log Replay Service (LRS) to migrate

Completed

Log Replay Service (LRS) is a tool that enables custom migrations of databases from on-premises SQL Servers to SQL Managed Instance in the cloud. It uses log shipping technology and is useful in cases where more control is needed, when there's little tolerance for downtime, or when Azure Data Migration Service can't be used.



LRS can be used directly with PowerShell, CLI cmdlets, or API, to manually build and orchestrate database migrations to SQL Managed Instance. Some of the reasons to consider using LRS include:

- More control over the database migration project
- Little tolerance for downtime on migration cutover
- Inability to install DMS executable in the environment
- Lack of file access to database backups
- Inability to open networking ports from the environment to Azure

Understand migration types

There are two migration modes available for LRS.

Mode	Description	Recommended for	Backup Chain Availability
Autocomplete	Automatically finishes the migration when the last backup file is restored	Passive workloads	Entire backup chain must be available in advance
Continuous	Continuously scans for new backup files and restores them, allowing for data catch-up	Active workloads	Backup chain can be added during migration

Regardless of the mode, plan to complete the migration within 30 days, as the LRS job will be automatically canceled after this time.

Secure the migration process

To run LRS, you must have one of the following Azure role-based access control (RBAC) role: Subscription Owner, SQL Managed Instance Contributor, or a custom role with the permission `Microsoft.Sql/managedInstances/databases/*`.

An Azure Blob Storage account is required and works as an intermediary storage for backup files between your SQL Server instance and your SQL Managed Instance. To use Azure Blob storage with a firewall, another configuration is required. You must add the SQL Managed Instance subnet to the storage account's virtual network firewall rules using MI subnet delegation and the Storage service endpoint. Also, you can use either a SAS token or a managed identity to access your Azure Blob Storage account, but not both.

Improve backup and restore performance

You can split full and differential backups into multiple files, instead of using a single file, to improve backup and restore performance. This is because multiple files can be read or written to in parallel, reducing the time it takes to complete the backup or restore operation.

Also, enabling backup compression can help improve network transfer speeds. Compressed backups are smaller in size, which means they take less time to transfer over the network. This can be especially useful when transferring large backups to or from Azure.

We strongly recommend enabling `CHECKSUM` for backups, even though it isn't required. SQL Managed Instance performs an integrity check on backups without `CHECKSUM`, which can increase the time it takes to restore the database. By enabling `CHECKSUM`, you can speed up the restore operations.

5. Migrate using Managed Instance link

Migrate using Managed Instance link

Completed

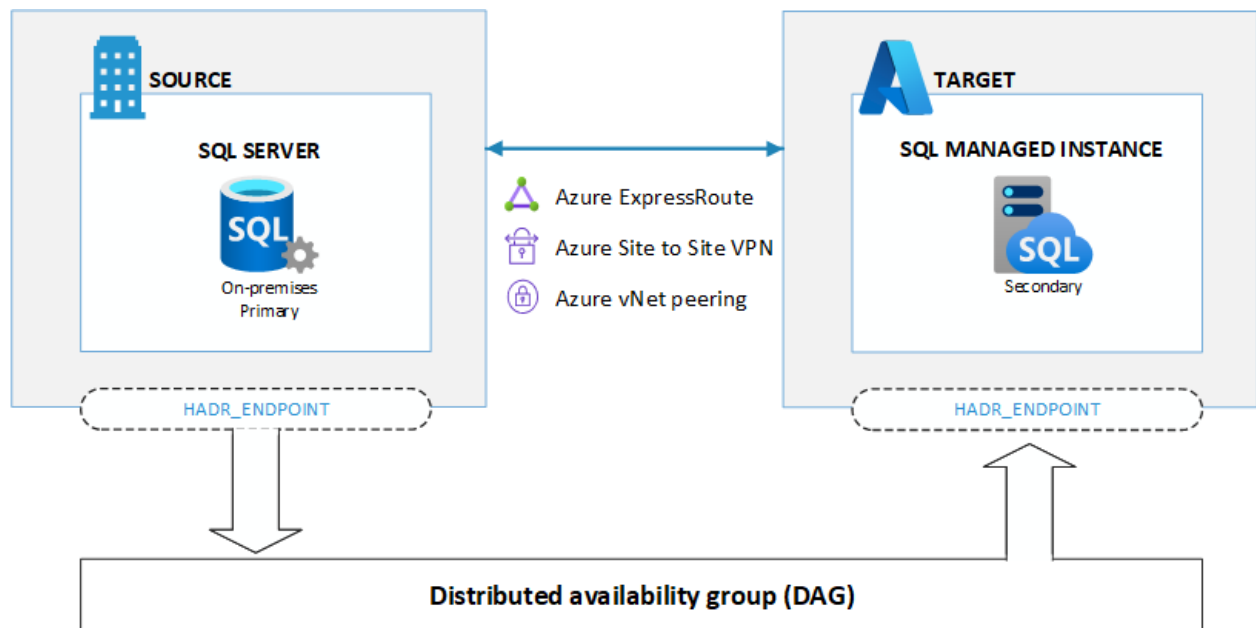
Azure SQL Managed Instance link feature offers a true online migration option compared to all other services and tools available. In addition, by partially running workloads on Azure, organizations can adopt a phased approach to cloud migration while still taking advantage of the benefits that Azure provides.

While the link is designed to replicate one database per link, it can be configured to replicate multiple databases from a single SQL Server instance to one or more SQL managed instances, or even replicate the same database to multiple SQL managed instances. This is achieved by setting up multiple links, each corresponding to a database-managed instance pair.

Hybrid flexibility with Azure SQL Managed Instance link feature

Azure SQL Managed Instance link feature allows you to replicate your SQL Server databases hosted anywhere to Azure, and fail over to the cloud in the event of a disaster or major business disruption. Azure SQL Managed Instance link also ensures seamless failover between the primary and secondary databases.

One of the advantages of using Azure SQL Managed Instance is that it's a platform as a service (PaaS), which means that the latest hardware maintenance, patching, and updates are applied and managed automatically by Azure. This ensures that your database environment is always up-to-date and secure, while also reducing the risk of downtime due to hardware failures or software vulnerabilities.



As we can see above, the link feature uses distributed availability group (DAG) and it's scoped per database (one link per one database). This allows you to consolidate multiple parallel SQL Server databases into an Azure SQL Managed Instance or scale them out across multiple instances and regions worldwide.

The link feature provides two types of replication:

- **One-way replication.** One-way replication is available for SQL Server versions 2016 and 2019 and allows you to replicate data one way from a SQL Server instance to your managed instance.
- **Two-way replication.** SQL Server 2022 provides a two-way replication feature, where you can replicate data between your managed instance and SQL Server instances, manually fail over during a disaster, and manually fail back after the disaster is mitigated. It supports an online failover but an offline failback. A preview of the online failback is available to sign up for.

Extended capabilities to the cloud

In addition to migrate workloads, there are several ways to use the link feature and use Azure services and resources, which include:

Feature	Description
Offload read-only workloads	You may want to configure secondary replicas on your SQL Server to Azure to offload reporting needs. The link feature is database scoped, allowing for consolidation of read-only workloads in Azure, which can be used to bring data closer to customers in any supported region worldwide with minimum effort.
Automated backups	Secondary replicas running on Azure SQL Managed Instance are automatically backed up to your Azure Blob Storage account, which significantly reduces administrative efforts and improve reliability.

Feature	Description
Business continuity	As a disaster recovery solution, the link feature allows you to fail over to Azure SQL Managed Instance and fail back after the disaster is mitigated.

Enable the link feature

To configure the link feature, you must follow the same steps regardless of whether you're migrating to Azure SQL Managed Instance, configuring disaster recovery on the cloud, offloading workloads to Azure, or aiming to reduce backup operations and management costs.

You can use either a wizard in SQL Server Management Studio (SSMS) or scripts. The main advantage of using scripts is that they can be automated, which can improve your deployment process, saving time and effort.

- Replicate a database by using [Azure SQL Managed Instance link wizard available in SSMS](#).
- Replicate a database by using [T-SQL and PowerShell scripts](#).

There are a few SQL Server features that aren't supported by Azure SQL Managed Instance link. For example, you can't enable the link feature if the functionality that's used on the primary SQL Server database isn't supported on Azure SQL Managed Instance, such as file tables and file streams.

For the full list of supported features, see [Limitations of Azure SQL Managed Instance link](#).

As we've seen, Azure SQL Managed Instance link feature enables organizations to confidently extend their SQL Server environments to Azure while also benefiting from the scalability, performance, and security features that Azure SQL Managed Instance offers.

Choosing between migration methods

The choice between Managed Instance link and LRS depends on your specific requirements:

Consideration	Managed Instance link	Log Replay Service
Minimum downtime required	Best choice - cutover in seconds	Longer cutover, especially for Business Critical tier
Need to read data during migration	Supported	Not available - database is in restoring state
SQL Server version	2016 and later	2012 and later
SQL Server edition	Enterprise, Standard, Developer	All editions
Network configuration	Requires VPN or private endpoint setup	Works with public endpoint by default
Migration duration	Unlimited	Maximum 30 days

Consideration	Managed Instance link	Log Replay Service
Reverse migration needed	Supported (depends on SQL MI update policy)	Not supported

For most business-critical workloads targeting either General Purpose or Business Critical SQL Managed Instance, the Managed Instance link provides the best migration experience with minimal downtime.

LRS is well-suited for general purpose workloads where some planned downtime is acceptable, or when migrating from older SQL Server versions or editions not supported by the Managed Instance link.

6. Move data to SQL Managed Instance

<https://learn.microsoft.com/en-us/training/modules/migrate-sql-workloads-azure-managed-instances/6-synchronize-data>

Move data to SQL Managed Instance

Completed

Many migrations involve a period when the on-premises and the cloud database must be kept synchronized. For example, there might be a time when clients make changes to both databases.

You've migrated the sports retail products database into Azure SQL Managed Instance. The website is already using the cloud database. You're starting to reconfigure clients to use the new database. You've decided to transition users to the new system in batches. For each team, you'll take time to resolve any problems before migrating the next users. This approach allows for troubleshooting and problem resolution without disrupting all users at once. Next, you'll reconfigure the data analysis system to use the new database in Azure. During this time, you want to ensure that the cloud and on-premises databases are synchronized every hour.

You'll explore various methods for implementing data synchronization. These methods can also be employed to selectively migrate data, should you require only a subset of the tables to be transferred. This flexibility allows for a more tailored approach to data migration.

Connectivity options with on-premises servers

Often, you want to keep data in on-premises databases synchronized with Azure SQL Managed Instance. You might want to stage the migration of client applications to the new database, for example, which means there's a period when clients connect to both databases.

Before you choose a data synchronization method, it's important to ensure you have connectivity that's secure. There are three different connectivity options available to establish communication between computers on-premises and resources in Azure.

- **Point-to-Site.** A Point-to-Site (P2S) VPN gateway connection lets you create a secure connection to your virtual network from an individual client computer.
- **Site-to-Site.** A Site-to-Site VPN gateway is used to connect an entire on-premises site to the Azure network.
- **ExpressRoute.** Azure ExpressRoute enables you to create private connections between Azure datacenters and on-premises infrastructure, or infrastructure in a colocation environment. ExpressRoute connections don't go over the public internet, and offer more reliability, faster speeds, lower latencies, and higher security than typical internet connections.

Public endpoint

Public endpoint for SQL Managed Instance helps you connect to the database from the internet without using a VPN, and is designed for data communication only. Public endpoint for data can simultaneously coexist with the private endpoint. For security reasons, the implementation allows for Separation of Duties (SoD) between a database administrator and a network administrator when enabling the public endpoint.

To enable public endpoint for managed instance, two steps are required. For SoD, you'll need two separate roles, with the following database and network permissions, to complete these steps:

1. A database administrator who has role-based access control permissions in scope `Microsoft.Sql/managedInstances/*` must run a PowerShell script to enable public endpoint for managed instance.
2. A network administrator who has role-based access control permissions in scope `Microsoft.Network/*` must open the port 3342 used by the public endpoint on the network security group (NSG), and provide a UDR route to avoid asymmetric routing.

Choose a synchronization method

You can use many methods to synchronize data from a SQL Database managed instance to an on-premises server and back.

Native backup and restore

You can restore a database in Azure SQL Managed Instance from an Azure Blob Storage file using Shared Access Signature (SAS).

This involves creating a credential with access to Azure Blob Storage, and then using the `BACKUP DATABASE` command with the `COPY_ONLY` option. If your database is larger than 200 GB, you can use a striped backup by providing several URL locations.

```
BACKUP DATABASE YourDatabase TO URL = 'https://youraccount.blob.core.windows.net/yc
```

To restore the database in SQL Managed Instance:

```
RESTORE DATABASE YourDatabase FROM URL = 'https://youraccount.blob.core.windows.net'
```

BACPAC file using SqlPackage

A BACPAC file is essentially a zipped version of both your database's metadata and data. While this deployment method is compatible with SQL Database, SQL Managed Instance doesn't support migration via BACPAC within the Azure portal. As an alternative, the [SQLPackage utility](#) should be used with the BACPAC file.

Bulk Copy Program (BCP)

The [BCP utility](#) is a command-line tool that exports tables to files so you can import them. Use this approach to migrate from a single SQL Database to SQL Managed Instance and back.

Azure Data Factory (ADF)

[Azure Data Factory](#) is built for data movement and orchestration, with the focus on ingestion. ADF has the integration runtime support to run SSIS packages, and the public internet support for SQL Managed Instance.

Transactional replication

[Transactional replication](#) is a way to move data between continuously connected database servers.

The process begins with a snapshot of the publication database objects and data. Once the initial snapshot is taken, any subsequent changes to the data or schema at the Publisher are typically delivered to Azure SQL Managed Instance in near real-time as they occur.

SQL Managed Instance is flexible because it can be a publisher, distributor, and subscriber.

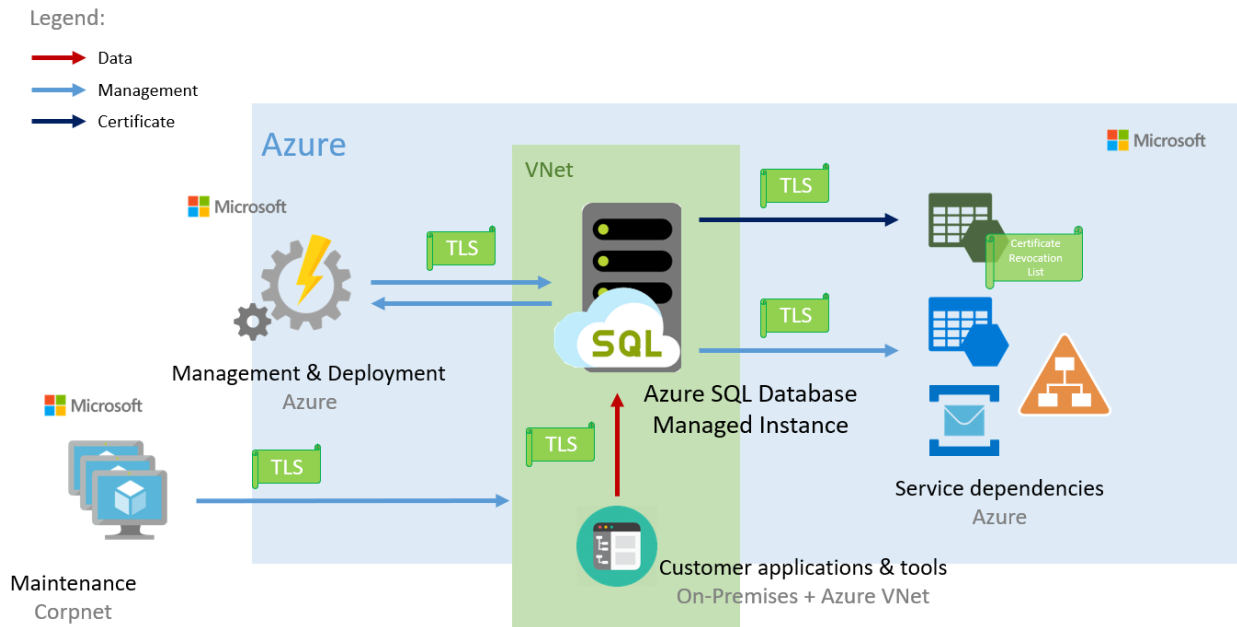
Replication is one of the few technologies that allows you to replicate parts of a table. We refer to these table parts as *articles*. This data is then sent to a distributor, which is a supplier of the data to any number of subscribers.

Requirements

- Connectivity uses SQL Authentication between replication participants.
- An Azure Storage Account share for the working directory used by replication.
- Open port 445 (TCP outbound) in the security rules of the managed instance subnet to access the Azure file share.
- Open port 1433 (TCP outbound) if the publisher or distributor is on a managed instance and the subscriber is on-premises.

Connecting applications to a SQL managed instance

A SQL managed instance must be placed inside an Azure virtual network subnet that's dedicated to managed instances. This deployment gives you a secure private IP address and the ability to connect to on-premises networks.



Users and client applications can connect to the managed instance database through the Azure portal, PowerShell, Azure CLI, and the REST API.

All communications are encrypted and signed using certificates. To check the trustworthiness of communicating parties, managed instances constantly verify these certificates through certificate revocation lists. If the certificates are revoked, the SQL managed instance closes the connections to protect the data.

7. Exercise - Migrate a SQL Server database to Azure SQL Managed Instance

<https://learn.microsoft.com/en-us/training/modules/migrate-sql-workloads-azure-managed-instances/7-exercise-migrate-sql-managed-instance>

Exercise - Migrate a SQL Server database to Azure SQL Managed Instance

Completed

Now it's your chance to migrate a SQL Server database using Log Replay Service.

In this exercise, you'll learn how to migrate a SQL Server database to Azure SQL Managed Instance using Log Replay Service.

Note

To complete this exercise, you will need an [Azure subscription](#).

Launch the exercise and follow the instructions.

[Launch Exercise](#)

8. Module assessment

<https://learn.microsoft.com/en-us/training/modules/migrate-sql-workloads-azure-managed-instances/8-knowledge-check>

Module assessment

Completed

9. Summary

<https://learn.microsoft.com/en-us/training/modules/migrate-sql-workloads-azure-managed-instances/9-summary>

Summary

Completed

In your sports clothing retail company, you needed an easy and quick way to migrate your products database into Azure. You chose Azure SQL Managed Instance because it provides a platform as a service (PaaS) implementation of SQL Server in the cloud. Azure SQL Database also has almost 100 percent compatibility with on-premises versions of SQL Server. You migrated the data, and ensured it was synchronized with the on-premises database so you could reconfigure client applications and the data analysis system.

Before SQL Managed Instance was available, system architects who wanted to migrate a database to Azure had two options:

- You could use Azure SQL Database, which provided a full PaaS solution with the consequent reduction in administrative time. However, this often required significant modification of the database, because Azure SQL Database doesn't support all the features of SQL Server.
- You could install SQL Server on a virtual machine in Azure. However, this infrastructure as a service (IaaS) solution requires you to run, administer, and update the virtual machine, its operating system, and SQL Server yourself. You could expect to spend more time on administration.

SQL Managed Instance provides a solution that's a full PaaS but supports almost all the features of SQL Server. This solution gives you the low administrative load you'd expect from Azure SQL Database but without the need to rewrite your database and change your client apps.

For additional reading, you can refer to the following URLs:

- [What is Azure SQL Managed Instance?](#)
- [Migrate SQL Server workloads to Azure SQL Database](#)
- [Tutorial: Migrate SQL Server to an Azure SQL Managed Instance online using DMS \(classic\)](#)