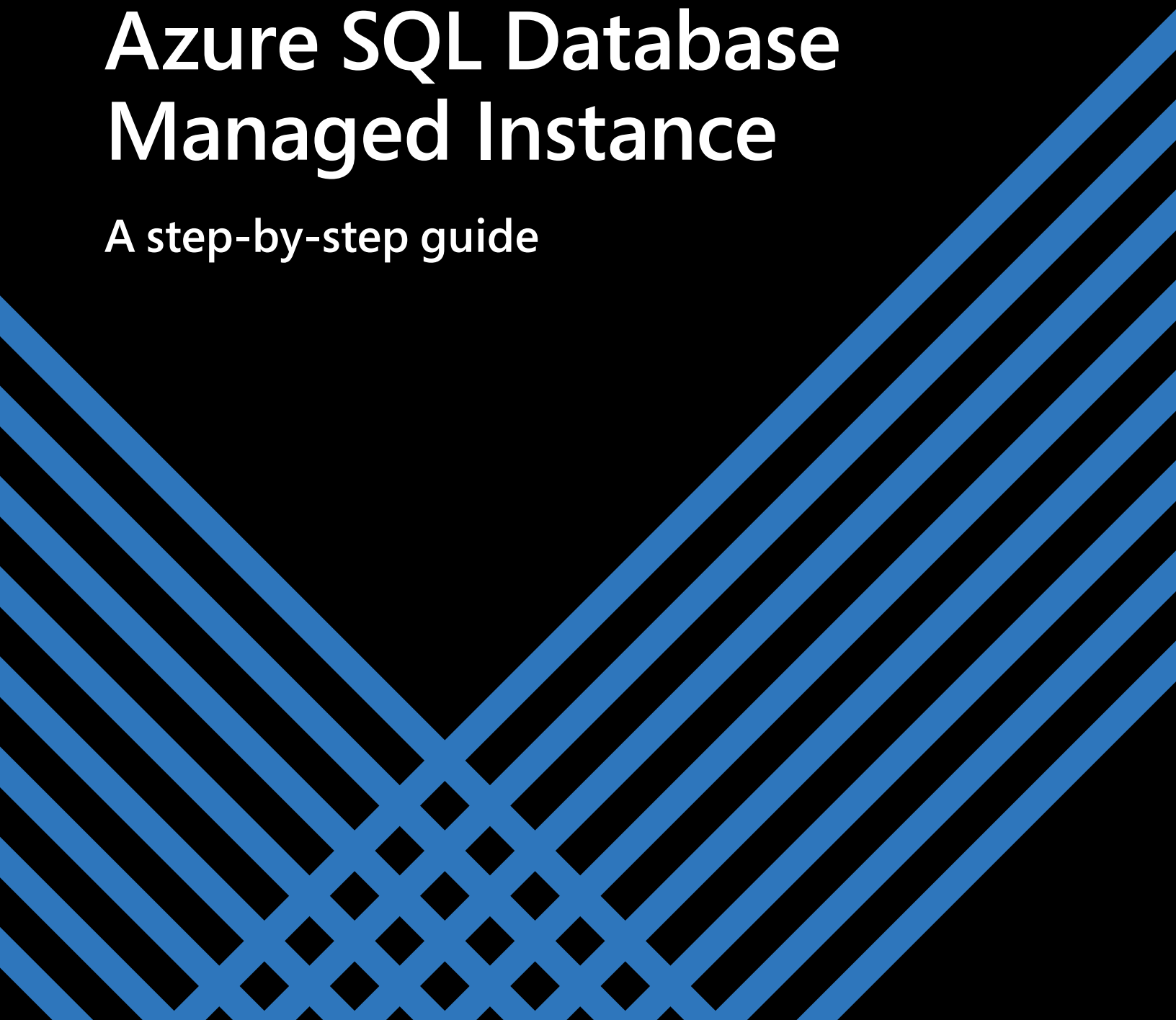


Migrating SQL Server to Azure SQL Database Managed Instance

A step-by-step guide



Who should read this e-book?

This e-book is for SQL Server administrators and system administrators tasked with migration of workloads that require thorough planning, critical thinking, and careful attention to detail. This e-book helps structure your tactical engagement by detailing the resources, workflows, and steps for migrating legacy Microsoft SQL Server (SQL Server 2008 and 2008 R2) workloads from on-premises to Microsoft Azure SQL Database Managed Instance.

Contents

01	When to migrate to Azure SQL Database Managed Instance	<u>01</u>
02	Assess	<u>03</u>
03	Migrate	<u>06</u>
04	Optimize	<u>15</u>
05	Get started	<u>18</u>

© 2018 Microsoft Corporation. All rights reserved. This document is provided "as is." Information and views expressed in this document, including URL and other internet website references, may change without notice. You bear the risk of using it. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

01

When to migrate to Azure SQL Database Managed Instance

The mix of databases in the cloud is growing significantly, and organizations need a durable framework to lift and shift databases to the cloud while extending what they already have. With end of support for Microsoft SQL Server 2008 and 2008 R2 coming on July 9, 2019, many organizations are migrating their legacy SQL Server databases to an appropriate cloud platform. But some important factors need to be considered before migration—like cost, compatibility, and post-optimization.

Azure SQL Database Managed Instance is a better solution for migration because it has familiar features, lowers your TCO, and is fully managed. It enables you to seamlessly lift and shift SQL Server applications to Azure without making changes to your code. And it provides a portfolio of migration tools that can help you choose the right mix of cloud and on-premises data for your business.

Migration to Azure SQL Database Managed Instance

You can accelerate your migration to the cloud while maximizing support for legacy SQL Server databases by taking advantage of an easy migration strategy from Microsoft. Azure SQL Database Managed Instance offers a simple phased and guided approach to rehost your apps to Azure with limited or zero application code changes.

Migrating SQL Server databases to Azure SQL Database Managed Instance consists of three simple steps—assess, migrate, and optimize (Figure 1). For each step, a set of integrated tools and offerings helps ensure that you can choose the scope of migration, from a single database to an entire datacenter.

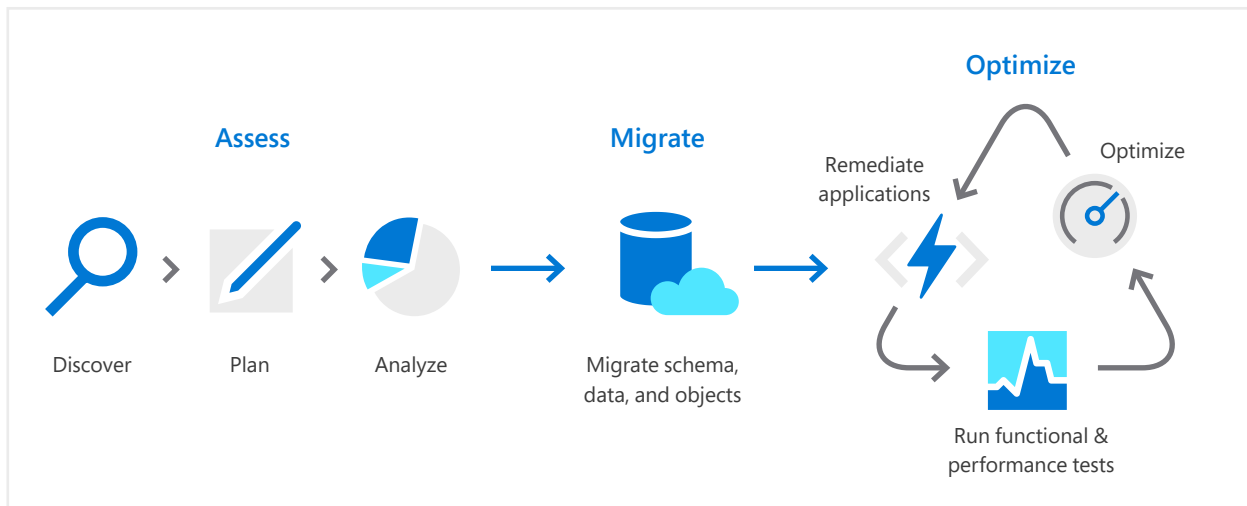


Figure 1: The three phases of migrating SQL Server data to Azure SQL Database Managed Instance

02

Assess

Discovering existing databases, and then planning and assessing the databases for cloud migration, is the first step of migration. To help you discover and understand the size of your environment for your data migration to the cloud, Microsoft provides the Data Migration Assistant.

Data Migration Assistant

Data Migration Assistant reviews the target migration environment and surfaces new features that will benefit your data post-migration. You can [download the latest version of Data Migration Assistant for free.](#)

To perform your assessment by using Data Migration Assistant:

1. In Data Migration Assistant, select the **New (+)** icon, and then select **Assessment** under Project type. Give your project a name, and then select **SQL Server** under Source server type and **Azure SQL Database** under Target server type.
2. Select **Check database compatibility** and **Check feature parity** for the assessment report type.
3. Select **Add Databases** to add databases to assess. Here you need to enter the source SQL Server instance name, choose the authentication type, and set the correct connection properties. You can also add databases from multiple SQL Server instances by using the **Add Sources** button.
4. Click **Next** to begin the assessment. The duration of the assessment depends on the number of selected databases and the size of each database and its schema. Results are displayed per database. Assessments for Azure SQL Database provide migration blocking issues and feature parity issues for the selected database. You can review and assess the results by selecting the appropriate specific options.

The **SQL Server feature parity** option displays a comprehensive set of recommendations, substitute approaches available in Azure, and mitigating steps.

The **Compatibility issues** option displays a list of partially supported or unsupported features that block migrating on-premises SQL Server databases to Azure SQL Database. It also offers recommendations on how to overcome those issues.

5. After all the assessments are done, you can also export the result to either a JSON or CSV file and analyze the data.


Microsoft Assessment and Planning (MAP) Toolkit

The Microsoft Assessment and Planning (MAP) Toolkit can help you discover all database assets and their characteristics, such as size and database details, as well as statistics on the number of tables, views, and stored procedures within those databases. You can use the result to right-size your database on Azure. The [MAP Toolkit](#) is available as a free download.

Additional requirements to assess

It's also important to analyze and plan for other requirements, such as performance, compliance, cost, security, and availability. For a secure connection, you need to plan to create and implement a virtual private network (VPN) or another network with end-to-end encryption. Azure SQL Database Managed Instance is locally available in the standard high-availability model. Three copies of your database are used to keep the data online and accessible during patching and transient hard failures. For disaster recovery, you can use geo-replication features to protect your database and application against wider regional failures.

For more information on planning, see [What is a managed instance?](#)



03

Migrate

Once you've completed discovery and assessed your environment, the next step is to migrate your databases from your existing on-premises database to the cloud. Azure offers migration tools that help you get to the cloud faster and more intelligently:

Azure Database Migration Service enables migration with near-zero downtime.

Native RESTORE from URL enables native backups from SQL Server, with some downtime.

Azure Data Migration Guide

The Azure Database Migration Guide can help you plan your migrations and provide information about migration tools that are available. Input your source and target platform choices into the guide, and it will create a customized playbook to suit your individual scenario. Get started with [exploring the guide](#) now.

Note: Database Migration Service does not yet support migrating SQL Server Integration Services (SSIS), SQL Server Analysis Services, or SQL Server Reporting Services (SSRS) components from SQL Server on-premises environments. For migrating SSIS, you can provision Azure-SSIS integration runtime (IR) in Azure Data Factory. For more information, see [Create the Azure-SSIS integration runtime in Azure Data Factory](#). For SQL Server Analysis Services, you can migrate to Azure Analysis Services.

For more information, see [Azure Analysis Services videos on Channel 9](#). SSRS currently has no cloud-based equivalent, but you can rewrite reports by using Microsoft Power BI or migrate reports to SSRS on an Azure virtual machine.

Migration with Azure Database Migration Service

Azure Database Migration Service is a fully managed migration service designed to enable seamless migrations from multiple database sources to Azure data platforms with minimal downtime, as shown in Figure 2 on the next page. Once you've performed a thorough assessment and resolved any issues in your source database (as identified by the Data Migration Assistant, discussed above), you can migrate a sample schema to Azure SQL Database Managed Instance by using Database Migration Service. These are the basic steps:

1. Provision and configure an Azure Virtual Network (VNet).
2. Create an optimally sized Azure SQL Database managed instance.
3. Create an Azure Storage account for backing up your source database.
4. Create a Database Migration Service migration project with the source and target defined.
5. Connect and test your application.

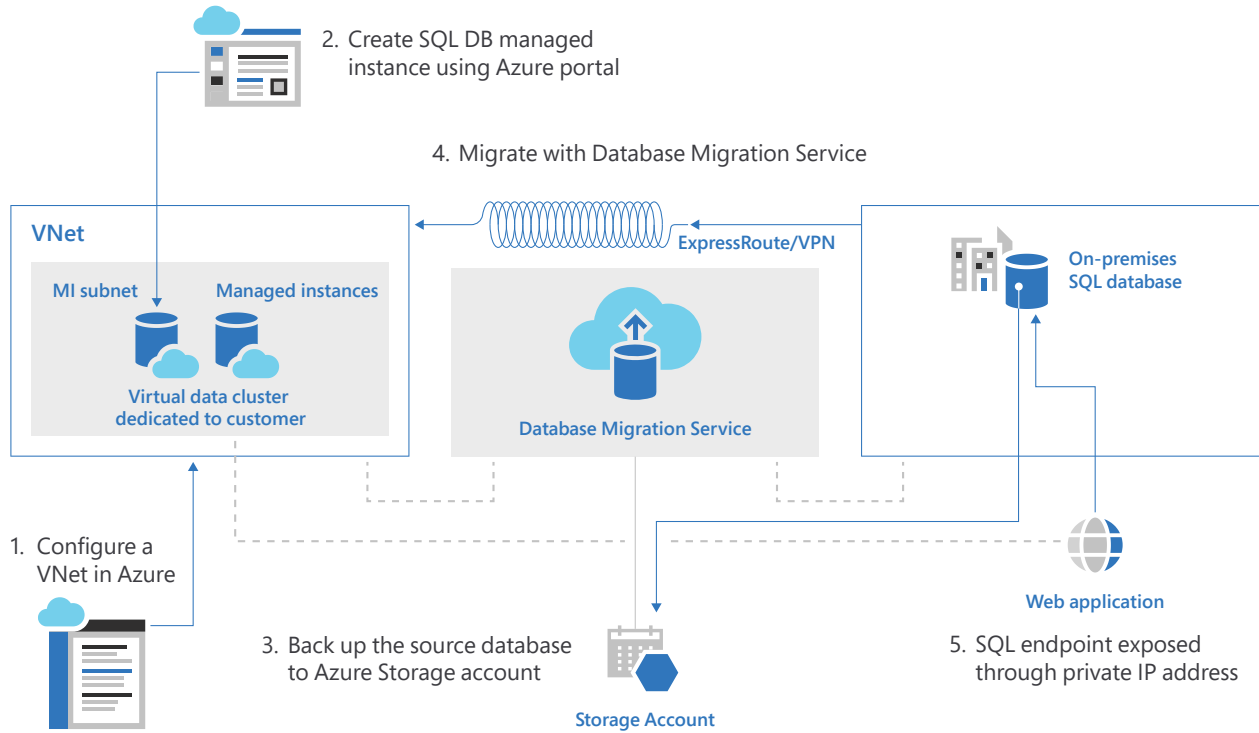


Figure 2: How Azure Database Migration Service works

Configure a VNet

These steps are described in more detail below.

1. Create the VNet for migration using the Azure Resource Manager deployment model, which provides site-to-site connectivity to your on-premises source server via ExpressRoute or VPN. You can use the Azure portal, Microsoft [PowerShell](#), or [Azure CLI](#) for this. Azure SQL Database Managed Instance must be deployed within a VNet. For more information on VNet configuration, see [Configure a VNet for Azure SQL Database Managed Instance](#).
2. Once your VNet is in place, make sure that the 443, 53, 9354, 445, and 12000 communication ports are allowed by your VNet's network security group. For details, see [Plan virtual networks](#).
3. Depending on your environment, you might also need to change your firewall's rules to make sure that Azure Database Migration Service can access the source database that you want to migrate. You might also need to use [SQL Server Configuration Manager](#) to [allow access for the database engine](#). Enable the default TCP/IP port 1433 to allow Database Migration Service to access the source SQL Server instance and, in case of multiple named SQL Server instances, enable the SQL Server Browser Service and UDP port 1434 through your firewalls.

Create an optimally sized Azure SQL Database managed instance

1. Next, create an Azure SQL Database managed instance using the Azure portal. Select the [Azure SQL Database Managed Instance \(preview\)](#) resource from the Azure portal to create the managed instance. You need to select your subscription and check whether your subscription is approved for the preview.

Note: Azure SQL Database Managed Instance is currently launched as a gated public preview that requires that subscriptions be approved. If your subscription is not approved, send a request. (Once the service is fully launched, you won't need to go through this process.)

2. Using the VNet you configured above, create and apply a route table to the managed instance subnet. The table allows your managed instance to communicate with Azure Management Services. This is needed because the managed instance is in your private virtual network. If it's not allowed to communicate with the Azure service that manages it, it becomes inaccessible.
3. You can now proceed with Azure SQL Database Managed Instance using the Azure portal. Locate Managed Instance, and then select [Azure SQL Database Managed Instance \(preview\)](#) and click **Create**.
4. Fill out the Managed Instance form (shown in Figure 3) with the requested information, including the VNet, resource group, and pricing. Several vCore options are available, so you need to select the one that matches your needs based on the assessment completed during the assess step. You can also activate Azure Hybrid Benefit via the Azure portal by confirming that you have enough active licenses with Software Assurance.

After creating the managed instance, you can connect it to a VM by creating a VM in the new subnet in the VNet. Alternatively, you can connect it with SQL Server Management Studio by using the host name.

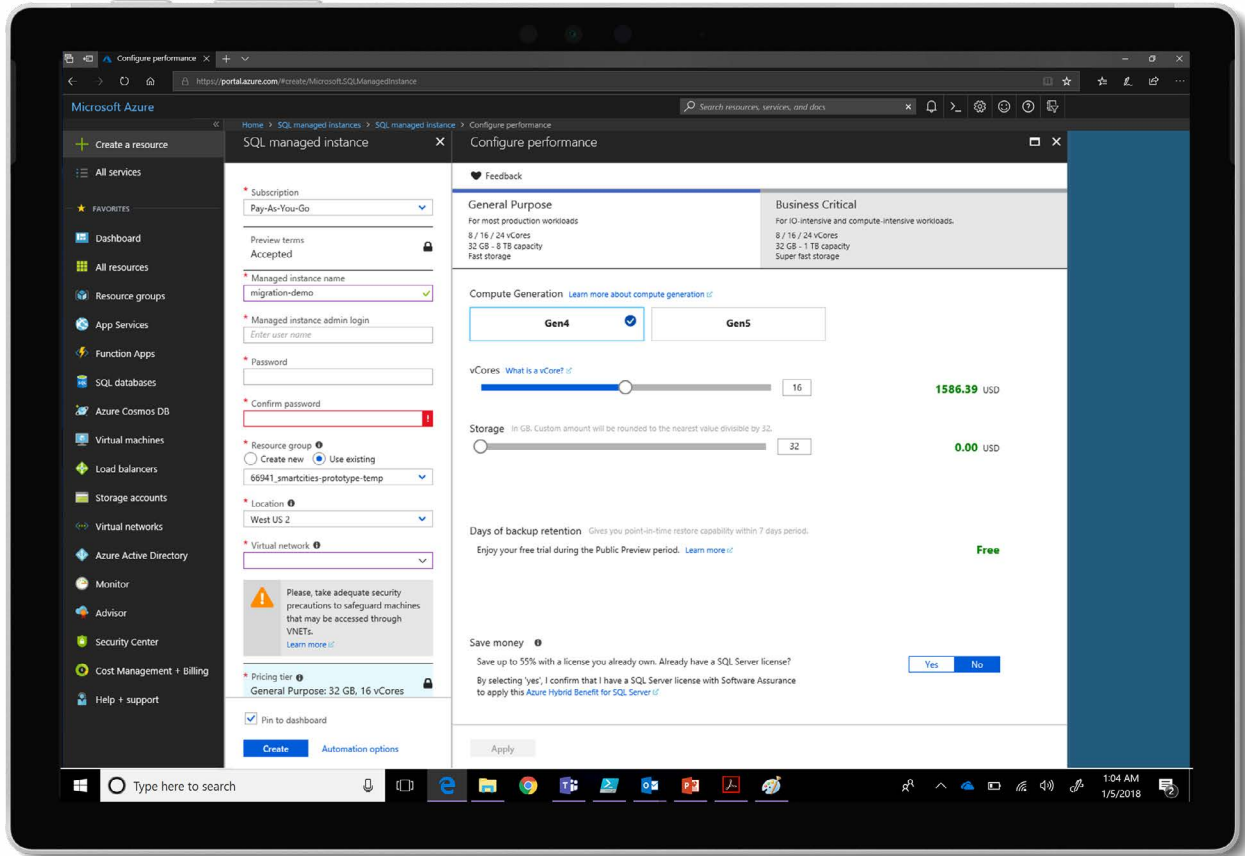


Figure 3: Managed Instance form

Create an Azure Storage account

Create an Azure Storage account as a network share to back up the source database during migration. Ensure that the service account running the source SQL Server instance has write privileges on the network share. Provide SAS details so the Azure Database Migration Service can access your storage account container to upload the backup files used for migrating databases to Azure SQL Database Managed Instance. After meeting all these prerequisites, you can migrate by using Database Migration Service.

Migrate with Database Migration Service

To create a Database Migration Service project, you first need to register the Microsoft.DataMigration resource provider with your subscription. For more information, see [Migrate SQL Server to Azure SQL Database Managed Instance offline using DMS](#). You can then create an Azure Database Migration Service instance. Here are the details:

1. To register your subscription with the Microsoft.DataMigration resource provider, select **Resource providers** in the subscription overview list. Search for migration, and then, to the right of Microsoft.DataMigration, select **Register**.
2. After registering, you can create the Azure Data Migration Service instance. From the Azure portal, select **Azure Database Migration Service**, and then click **Create**. On the **Create Migration Service** screen, specify a name for the service, the subscription, whether you're migrating a new or existing resource group, a pricing tier, and the VNet in which to create the service. For the pricing tier, you can assign more vCores to the service. This provides faster migrations to meet your intended timeline, but at the expense of added cost.
3. In the Azure portal, under the Azure Database Migration Service, select the instance that you created. Select **New Migration Project**. Specify a name for the project. In the **Source server type box**, select **SQL Server 2008 R2**. In the **Target server type box**, select **Azure SQL Database Managed Instance**. Select **Create**.
4. On the **Source Detail** screen, specify the connection details for the source SQL Server instance. If you have not installed a trusted certificate on the server, you also need to select a trusted certificate on your server. Save these settings.
5. On the **Select source databases** screen, select the database for migration. Select **Save**.
6. Next, you need to provide details about the target instance. On the **Target details** screen, specify the connection details for the target, which is the pre-provisioned Azure SQL Database managed instance to which the source database will be migrated. On the **Project summary** screen, review and verify the details associated with the migration project.

7. After verifying the details, select the recently saved project, and then select **Run migration**. Provide source and target credentials on the screen that appears.
8. Next, select the source database that you want to migrate. On the **Configure migration settings** screen, provide the FQDN or IP addresses of the server on the network share, a user name and password, and the SAS URI for the Azure storage container where the files will be uploaded.
9. Select **Save**, and then, on the **Migration summary** screen, in the **Activity name** box, specify a name for the migration activity. After verifying the details on the summary page, select **Run migration**.
10. To monitor the migration, on the migration activity screen, select **Refresh** to update the display until you see that the status of the migration shows as Completed.
11. After the migration is complete, you can select **Download report** to get a report listing the details associated with the migration process. Verify the target database in the Azure SQL Database Managed Instance environment.

That's how Azure Database Migration Service simplifies the entire end-to-end migration process.

Note: Restore of system databases is not supported. To migrate instance-level objects (stored in master or msdb databases), we recommend that you script them out and run T-SQL scripts on the destination instance.

Connecting and testing the application

During Azure SQL Database Managed Instance service provisioning, you choose the virtual network (VNet) and the subnet to achieve full networking isolation for your managed instances. Once created, instances in the VNet can be reached via Azure networking mechanisms (VPN and ExpressRoute gateways). The endpoint is exposed only through a private IP address. This allows safe connectivity from private Azure or hybrid networks. You can specify the private IP address at the time of creating the virtual network.

For more information, see [Connect your application to Azure SQL Database Managed Instance](#).

Native RESTORE from URL

Native RESTORE from URL is an alternative to using Database Migration Service for migration. A native SQL backup of the on-premises SQL Server databases can be created, put in Azure Blob storage, and restored straight to Azure SQL Database Managed Instance. Here's a code example:

```
BACKUP DATABASE <<yoursourcedatabase>> TO URL =  
'https://<storage_account_name>.blob.core.windows.  
net/<container>/yoursourcedatabase.bak'  
  
RESTORE DATABASE <<yoursourcedatabase>> FROM URL =  
'https://<storage_account_name>.blob.core.windows.  
net/<container>/yoursourcedatabase.bak''
```


04

Optimize

While moving SQL databases to Azure is the first step for your organization to modernize business processes, continual assessment of the migrated databases is also critical. It's important to confirm that the existing modeling was correct and to monitor the managed instance as loads increase and decrease to ensure ideal cost and performance.

Performance optimization

Run the performance validation queries against both the source and the Azure SQL Database Managed Instance databases, and then analyze and compare the results. Below are some performance-tuning scenario examples that you might encounter after migration:

Query regressions due to migration

When you migrate from an older on-premises database to Azure SQL Database, the database compatibility level changes, which results in a risk of performance regression. To reduce this risk, change the database compatibility level only after performance monitoring. First you need to change the [database compatibility level](#) to the level of the source version, and then use Query Store to get information about workload performance before and after the database compatibility level changes. Then move to the latest database compatibility level, and expose your workload to the latest query optimizer changes to let it potentially create new plans.

Missing indexes

Incorrect or missing indexes can cause extra I/O that leads to wasted memory and CPU. Use the graphical execution plan for any missing index references, and implement indexing suggestions generated by the [Database Engine Tuning Advisor](#). Use the [Missing Indexes DMV](#) or the [SQL Server Performance Dashboard](#).

Backup optimization

With Azure SQL Database Managed Instance, you don't need to create backups because it performs backups for you automatically. You can restore to any point in time within the retention period using [Point in Time Restore \(PITR\)](#).

Optimize and right-size your databases

Azure SQL Database Managed Instance introduces a pricing model based on vCores, which enables DBAs to select the right level of resources (storage and compute)—allowing you to right-size and optimize your destination environment in the cloud.

Security optimization

To further strengthen and optimize your database security in the cloud, Azure SQL Database Managed Instance offers various security features, including database-level authentication via Azure Active Directory, monitoring of database activities via auditing and threat detection, and access control of sensitive and privileged data through Transport Layer Security and Row-Level Security.

Azure Active Directory

You can configure and manage Azure Active Directory (Azure AD) authentication with Azure SQL Database Managed Instance to centrally manage database users and simplify permission management. Managed Instance requires permissions to read Azure AD for activities such as user authentication through security using the Azure portal and PowerShell. You can use Azure AD [Multi-Factor Authentication \(MFA\)](#) to increase data and application security while supporting a single sign-on (SSO) process.

Auditing

Auditing in Azure SQL Database Managed Instance tracks database server events and records them in an audit log. SQL Audit works at the server level and stores .xel log files in the Azure Blob account. This is the syntax for creating a server audit:

```
CREATE SERVER AUDIT [<your_audit_name>] TO URL (PATH
='<container_url>' [RETENTION_DAYS = integer])
GO
```

Securing your data

Azure SQL Database Managed Instance helps secure your data by providing encryption for data in motion via Transport Layer Security, which also provides protection for sensitive data in flight, at rest, and during query processing via Always Encrypted. For more details, see [Always Encrypted](#).

You can use Dynamic Data Masking to limit exposure to sensitive data by masking it from non-privileged users and helping to prevent unauthorized access to it. For further details, see [Dynamic Data Masking](#). Row-Level Security (RLS) helps you to control access to rows in a database table based on the characteristics of the user executing a query. For further details, see [Row-Level Security](#).

Azure SQL Database Threat Detection detects anomalous activities, flagging unusual or potentially harmful breaches. You can enable detection on the Threat Detection configuration page in the settings for your SQL Server managed instance. For more details, see [Azure SQL Database Managed Instance Threat Detection](#).

05

Get started

Azure SQL Database Managed Instance is the ideal solution for an easier lift-and-shift migration of your legacy SQL Server databases to the cloud. Azure services help IT teams migrate data to Azure SQL Database Managed Instance quickly and economically, with virtually no application design changes. Various tools offer full coverage for discovery, migration, and continual optimization while delivering flexibility to define your own migration pace. Get started—and finished—much faster than you thought possible with Azure SQL Database Managed Instance.

Prepare for [SQL Server 2008 and SQL Server 2008 R2 end of support](#)

Sign up for an [Azure free trial](#)—\$200 credit for 30 days, 12 months of free services

Get step-by-step guidance from the [Azure Database Migration Guide](#)

For details on Azure Data Migration Assistant, see [Perform a SQL Server migration assessment](#)



Resources

For more information about creating a VPN on Azure, see [Configure a VNet for Azure SQL Database Managed Instance](#).

For documentation on how to create a managed instance and restore a database from a backup file, see [Create an Azure SQL Database managed instance](#).

To learn about using the Azure Database Migration Service, see [Migrate SQL Server to Azure SQL Database Managed Instance using DMS](#).

For more on the performance and pricing of Azure SQL Database Managed Instance, see the [pricing](#) and [Azure Hybrid Benefit](#) pages.

For a list of countries/regions currently available for Azure SQL Database Managed Instance, see [Migrate your databases to a fully managed service with Azure SQL Database Managed Instance](#).

For the latest blog posts on these topics, visit the [Microsoft Data Migration Blog](#).