

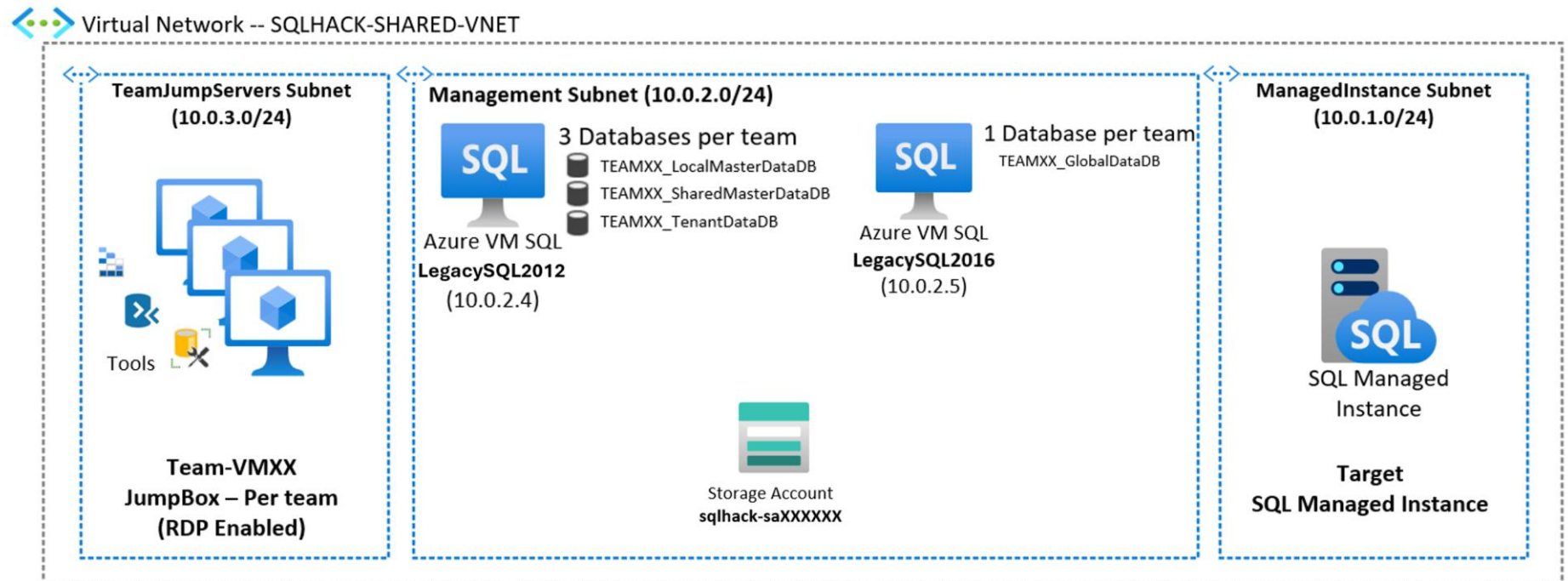
Database Migration Lab Step-by-step (Using Azure Data Studio)

V3.0

Contents

Migration architecture and Azure components	2
Generic Migration Content	3
1. Get the SQL Managed Instance FQDN	4
2. Assess the application databases for Azure SQL Database suitability using the Database Migration Assistant (DMA)	6
3. Migrate the application databases to Azure SQL Database managed instance using the Azure Data Studio (ADS) with migration extension and identify target Azure SQL SKU	19
4. Confirm application databases have been migrated to Azure SQL Managed Instance.....	39

Migration architecture and Azure components



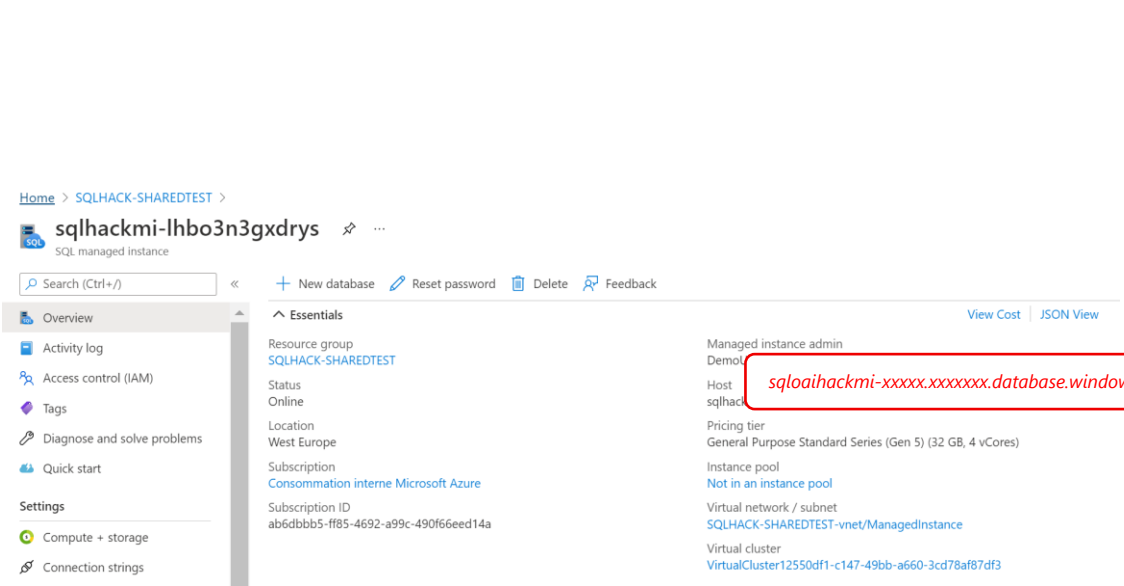
SQLOAIHACK-SHARED-VNET Single Virtual Network containing all workshop resources		
“TeamJumpServers” Subnet Each team is assigned a Win10 VM that mimics their company desktop	Management Subnet Several machines and services are already deployed within a dedicated subnet within the Virtual Network	“ManagedInstance” Subnet The Azure SQL Managed Instance has been deployed into a dedicated Subnet

Generic Migration Content

Narrative	Notes
<p><i>Notes for outside of the workshop:</i></p> <p><i>Familiarise yourself with Microsoft migration tools and the Azure Database Migration Guide</i></p>	<p>Azure Database Migration Guide: https://www.microsoft.com/en-us/download/default.aspx</p> <p>DMA & download link: https://docs.microsoft.com/en-us/sql/dma/dma-overview?view=sql-server-ver15</p> <p>Azure Data Studio and Migration Extension download Links: Download and install Azure Data Studio - Azure Data Studio Microsoft Learn Azure SQL migration extension for Azure Data Studio - Azure Data Studio Microsoft Learn</p> <p>Microsoft Migration Portal: https://datamigration.microsoft.com/</p> <p>Identify the right Azure SQL Database, Azure SQL Managed Instance or SQL Server on Azure VM SKU for your on-premises database https://docs.microsoft.com/en-us/sql/dma/dma-sku-recommend-sql-db?view=sql-server-ver15</p>

1. Get the SQL Managed Instance FQDN

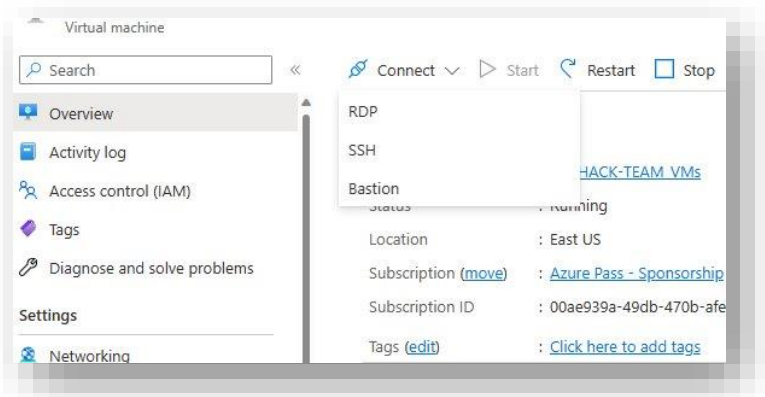
In this section we'll connect to the Azure Portal and retrieve SQL MI information: FQDN, ...

Narrative	Screenshot	Notes
<p>On your Win10 VM open Edge browser and got to the Azure Portal:</p> <p>HTTPS://portal.azure.com</p> <p>Log in with your Username and Password: (see <i>your Teams Group</i>)</p> <p>Similar to...</p> <p>User: sqloihackuserXX@M365x04034558.onmicrosoft.com</p> <p>Password: &sqloihack@demo_XX!</p> <p>In the Azure portal, open the “SQL0IAHACK-SHARED” Resource Group and locate the SQL managed instance and open it.</p> <p>Note (to a notepad) the Host Name (FQDN) in the format: sqloihackmi-xxxxx.xxxxxxx.database.windows.net</p> <p>Host should be: sqlhackmi-j754o5hum2r36.7a59bf01d694.database.windows.net</p>	 <p>The screenshot shows the Azure Portal interface for a SQL Managed Instance. The breadcrumb navigation at the top indicates the path: Home > SQLHACK-SHAREDTEST > sqlhackmi-lhbo3n3gxdrys. The instance name is 'sqlhackmi-lhbo3n3gxdrys' and it is of type 'SQL managed instance'. The left-hand navigation pane shows various options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Quick start, Settings, Compute + storage, and Connection strings. The 'Overview' section is expanded, showing details such as Resource group (SQLHACK-SHAREDTEST), Status (Online), Location (West Europe), Subscription (Consumption interne Microsoft Azure), and Subscription ID (ab6dbbb5-ff85-4692-a99c-490f66eed14a). On the right, the 'Managed instance admin' section is visible, with the 'Host' field highlighted by a red rectangular box. The host name displayed is 'sqloihackmi-xxxxx.xxxxxxx.database.windows.net'.</p>	

All other details from the “DB Migration Lab and Parameters.pdf”		
--	--	--

2. Assess the application databases for Azure SQL Database suitability using the Database Migration Assistant (DMA)

In this section we will use the Data Migration Assistant (DMA) to assess the applications database for suitability for migration to Azure Cloud.

Narrative	Screenshot	Notes
<p>Go to the Azure Portal and navigate to your dedicated Team VM.</p> <p>Click on Connect > Bastion.</p> <p>Before connecting, choose German as keyboard language.</p> <p>Then use the following credentials (copy to notepad if needed):</p> <p><i>Username:</i></p> <p>DemoUser</p> <p><i>Password:</i></p> <p>Demo@pass1234567</p>		

Using Bastion: **TEAM_VMs-BA**, Provisioning State: **Succeeded**

Please enter username and password to your virtual machine to connect using Bastion.

^ Connection Settings

Keyboard Language ⓘ
German

Username ⓘ
DemoUser

Authentication Type ⓘ
Password

Password ⓘ

Show

☒ Open in new browser tab

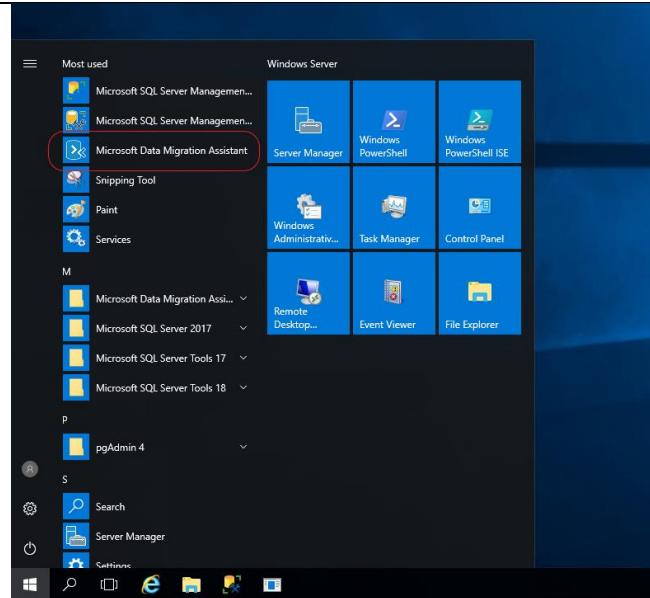
Connect

Azure OpenAI x Databases Hack

You should have a remote (Bastion) session open to your teams Win10 Management VM.

Now run the **Data Migration Assistant** (DMA) from the Start menu or Desktop icon.

We need to determine the suitability of the database(s) for migration to Azure. This includes checking for compatibility and feature support with Azure Database.

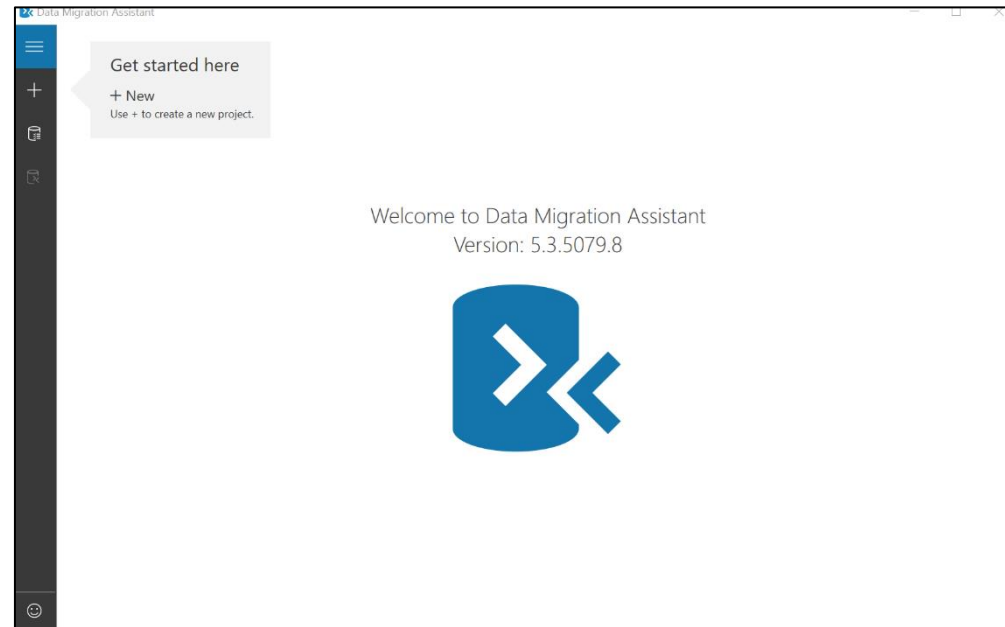


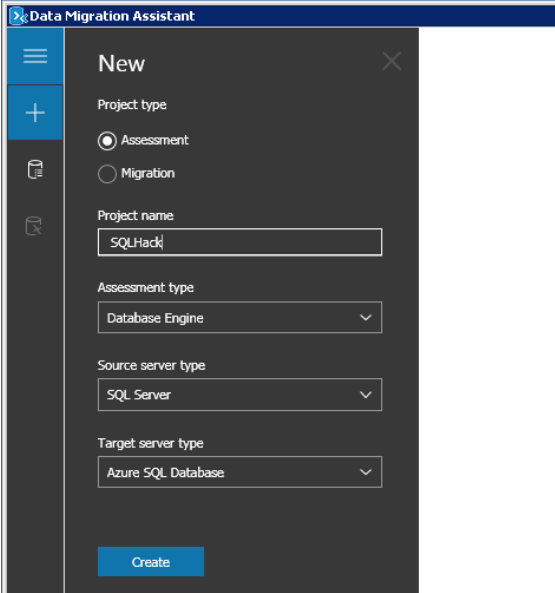
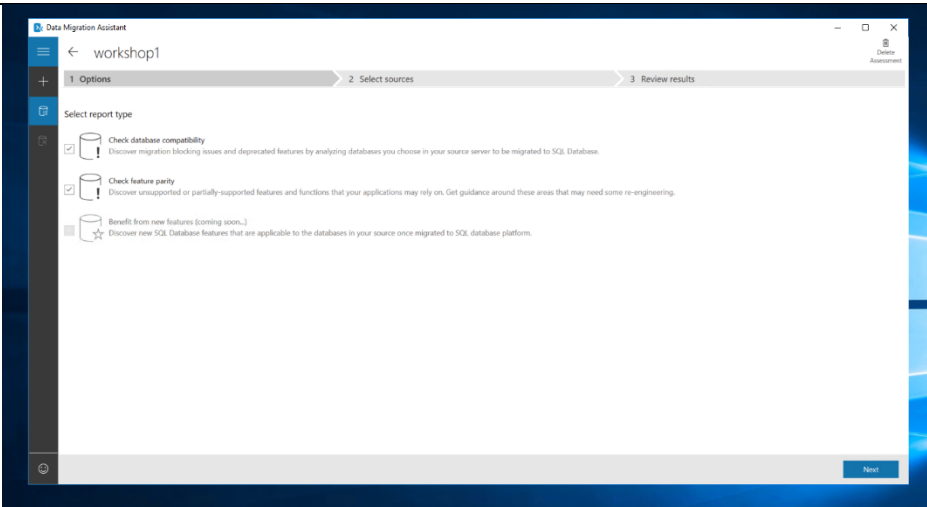
Database Migration Assistant (DMA) is a free download from Microsoft. It can be used to assess a number of database migration & upgrade scenarios not just SQL Server to Azure SQL Database.

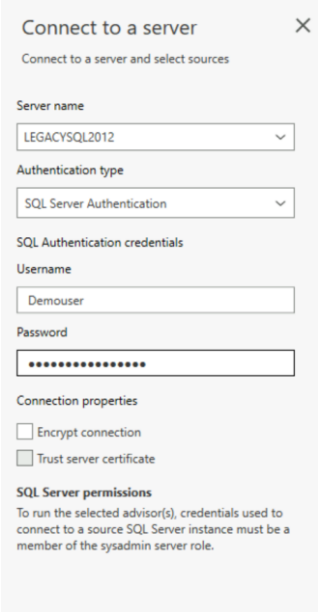
Azure OpenAI x Databases Hack

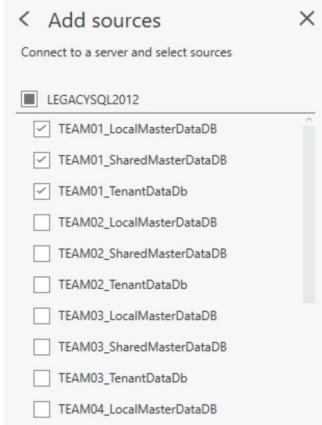
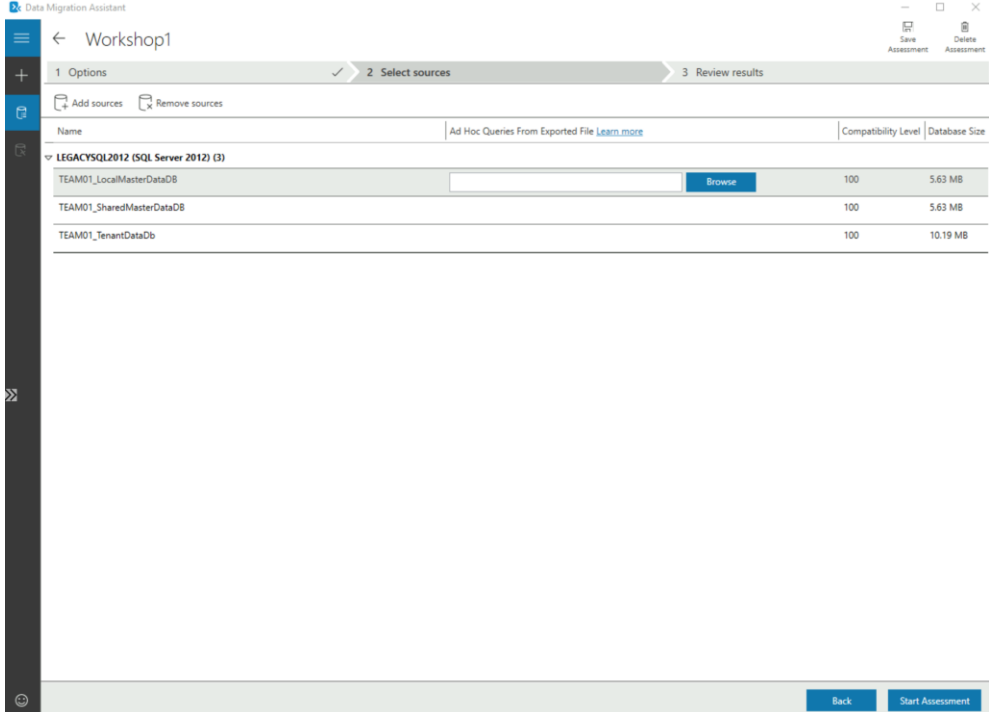
You should see this window open up:

Select the “+” button to create a **new assessment project**.



<p>Select/enter the following details in the mask:</p> <p>Project name: <i>Workshop1</i></p> <p>Assessment type: <i>Database Engine</i></p> <p>Source server type: <i>SQL Server</i></p> <p>Target server type: <i>Azure SQL Database</i></p> <p>Click 'Create'.</p>		<p>Our first project assessment assumes we will be migrating to Azure SQL DB, so the options shown in the screenshot need to be selected.</p>
<p>Select the assessment checks (Report Type) to be made:</p> <ul style="list-style-type: none"> • Check database compatibility • Check feature parity <p>Click 'Next'.</p>		<p>DMA can test for both database compatibility and feature parity compliance against the Azure target.</p> <p>As this is the initial evaluation, we are assessing a database(s) we will perform all of these tests.</p>

<p>Enter the source/legacy SQL details on the right side:</p> <p>Server Name: <i>LEGACYSQL2012</i></p> <p>Authentication Type: <i>SQL Server Authentication</i></p> <p>NOTE: Double check that there are no empty spaces in username or password!</p> <p>Username: <i>Demouser</i></p> <p>Password: <i>Demo@pass1234567</i></p> <p>Untick “Encrypt connection”</p> <p>Click ‘Connect’</p> <p><i>If you get an error logging in check the Win10 keyboard language</i></p>		<p>When performing this within your own subscription you will enter the host, authentication and connection types according to your company guidelines and practices.</p> <p><i>Bear in mind that DMA needs to connect to a source SQL Server using an account that belongs to the sysadmin role.</i></p> <p>As this document is produced within a workshop environment Active Directory, Certificates and encryption has not been setup.</p>
---	--	--

<p>Select only the 3 databases used by your 'Online Transaction Monitor' app. These will have a TEAMxx prefix where XX should be replaced by your team number.</p> <p>TEAMxx_LocalMasterDataDb</p> <p>TEAMxx_SharedMasterDb</p> <p>TEAMxx_TenantDataDb</p> <p>Click 'Add' to add them to the assessment.</p>		<p>DMA will show all databases located on the Source host and display them so you can decide which ones to include in this assessment project.</p> <p>Note that you can assess multiple databases at the same time.</p>
<p>You should now see the screen on the right with the 3 relevant TEAMxx databases listed.</p> <p>Select 'Start Assessment'</p>		<p>Note: DMA allows you to either 'Add' or 'Remove' additional data sources as needed at this point.</p> <p>Also note that DMA provides some high-level metadata about the databases including their compatibility level the total size of each database.</p> <p>Using Data Migration Assistant to assess an application's data access layer</p>

DMA will now show the results of the assessment using 2 separate reports, shown on the left side under “options”:

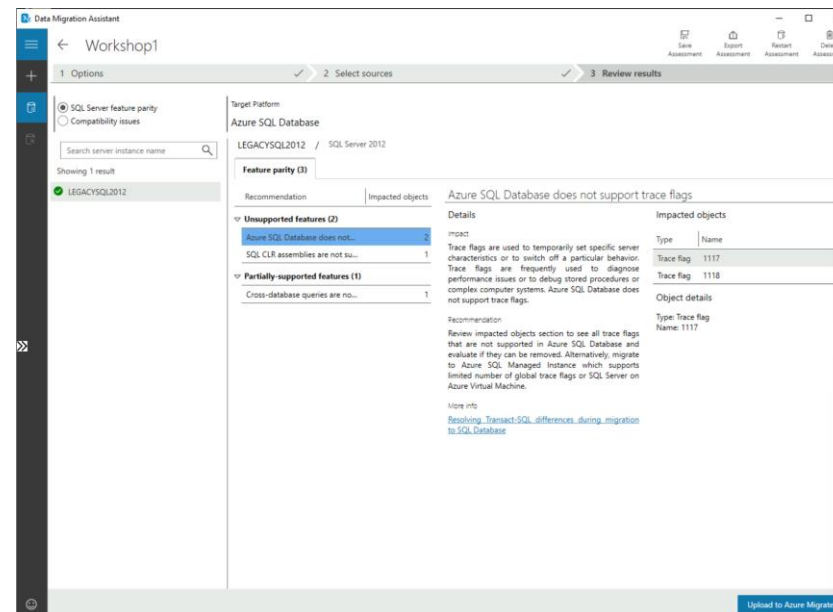
1. Report:

‘SQL Server feature parity’ which is a server level report highlighting any server settings or components (e.g. MSDTC) that the source DBs are using that isn’t supported on the target – in this case Azure SQL Database.

In our assessment there are **‘Unsupported’** or **‘Partially Supported’** features reported (**CLR, cross database queries, several trace flags**).

Switch to the 2. Report: **‘Compatibility Issues’** which is a database level report detailing individual objects that have compatibility issues.

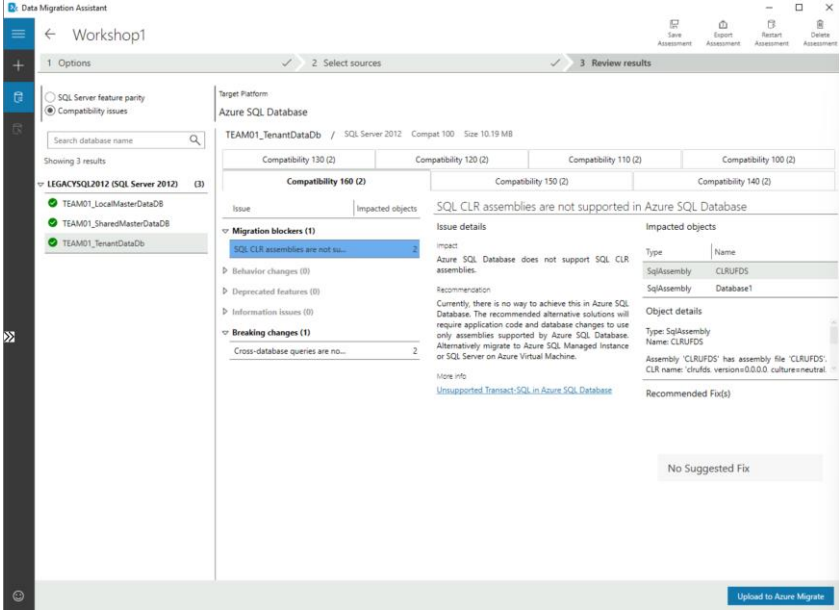
Select **‘TEAMxx_TenantDataDb’** on the left side. Note the **‘Migration blockers’** and **“Breaking Changes”** including CLR which the database uses.



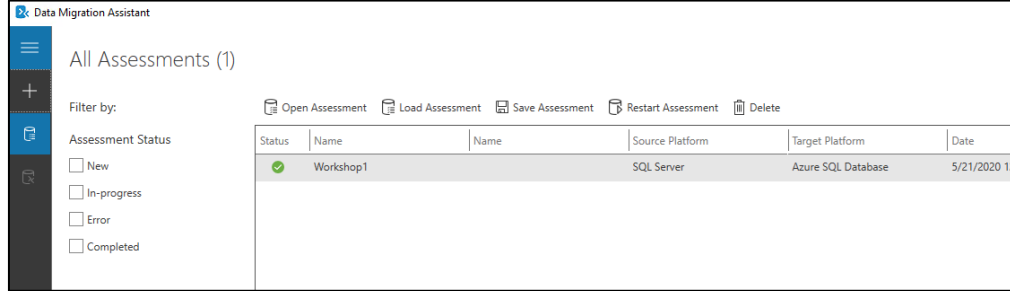
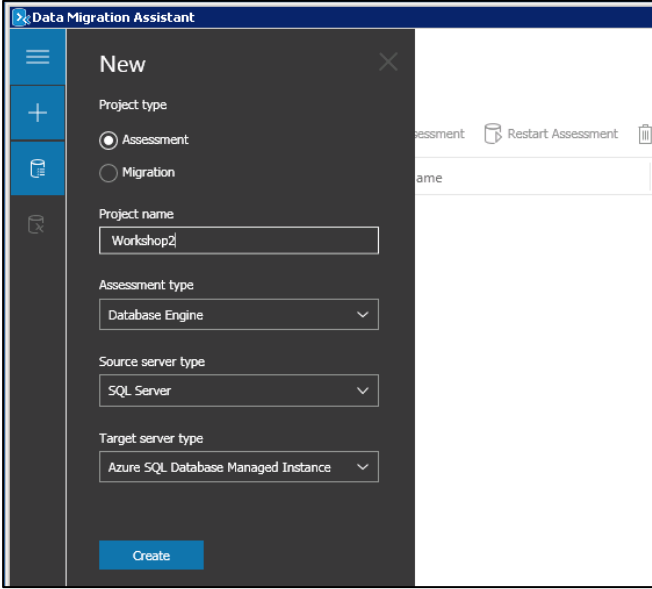
Note: Toggle the parity and compatibility issues radio button (top left) to switch between the 2 reports.

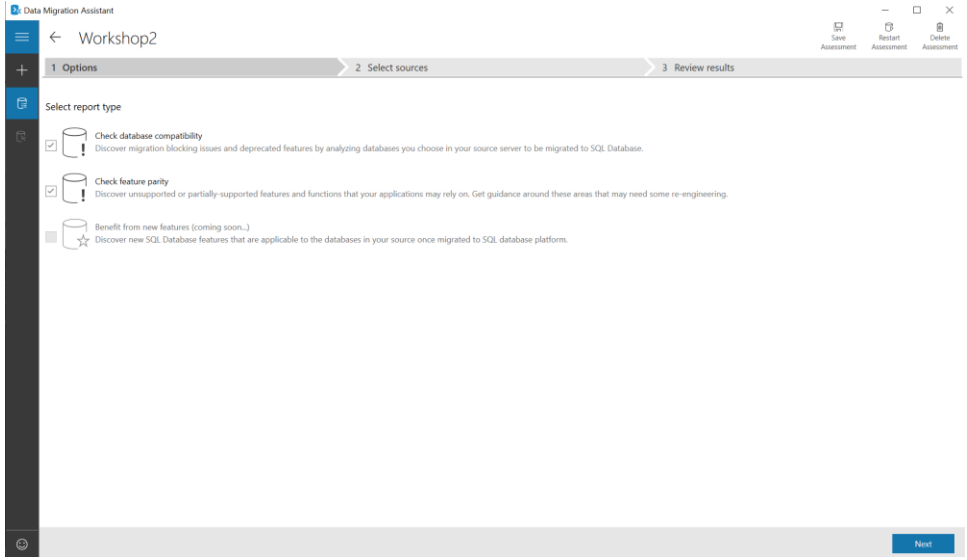
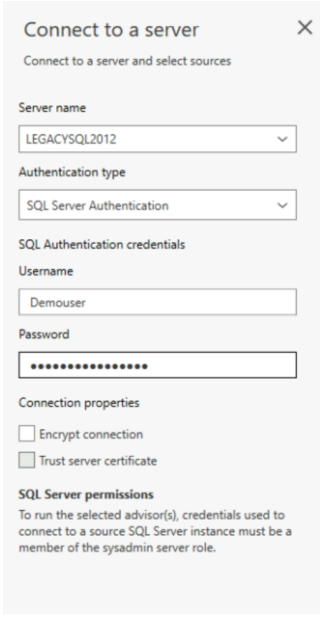
‘SQL Server feature parity’ shows what features are not supported in the target data source. Under the **‘Details’** and **‘Databases’** sections on the right you will find remedial action that are required and the databases impacted.

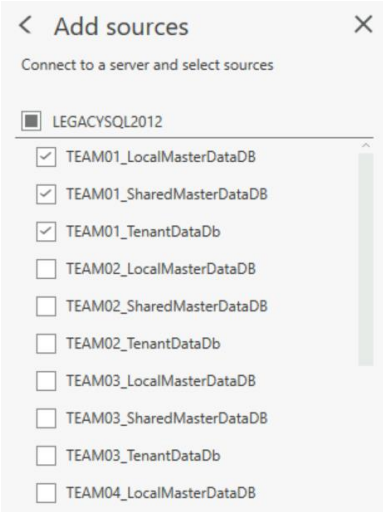
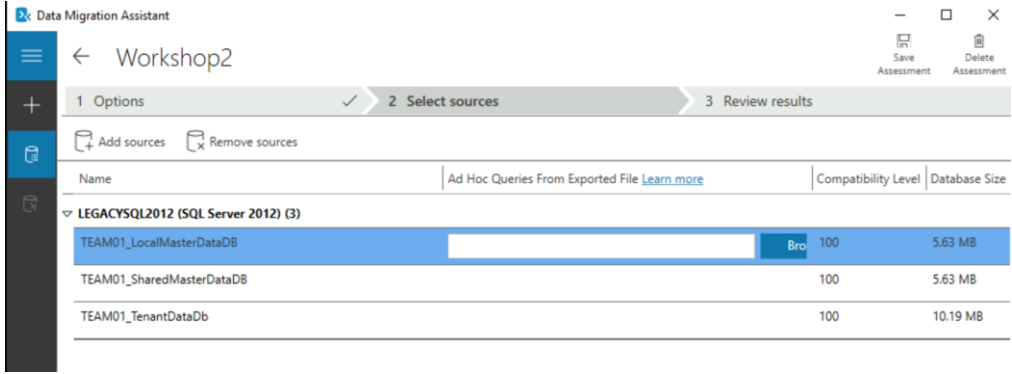
‘Compatibility Issues’ shows, over the compatibility tabs, issues that need to be addressed to permit the database(s) to run, in the chosen compatibility level (e.g. 160, 150, 140, 130, 120, 110, 100).

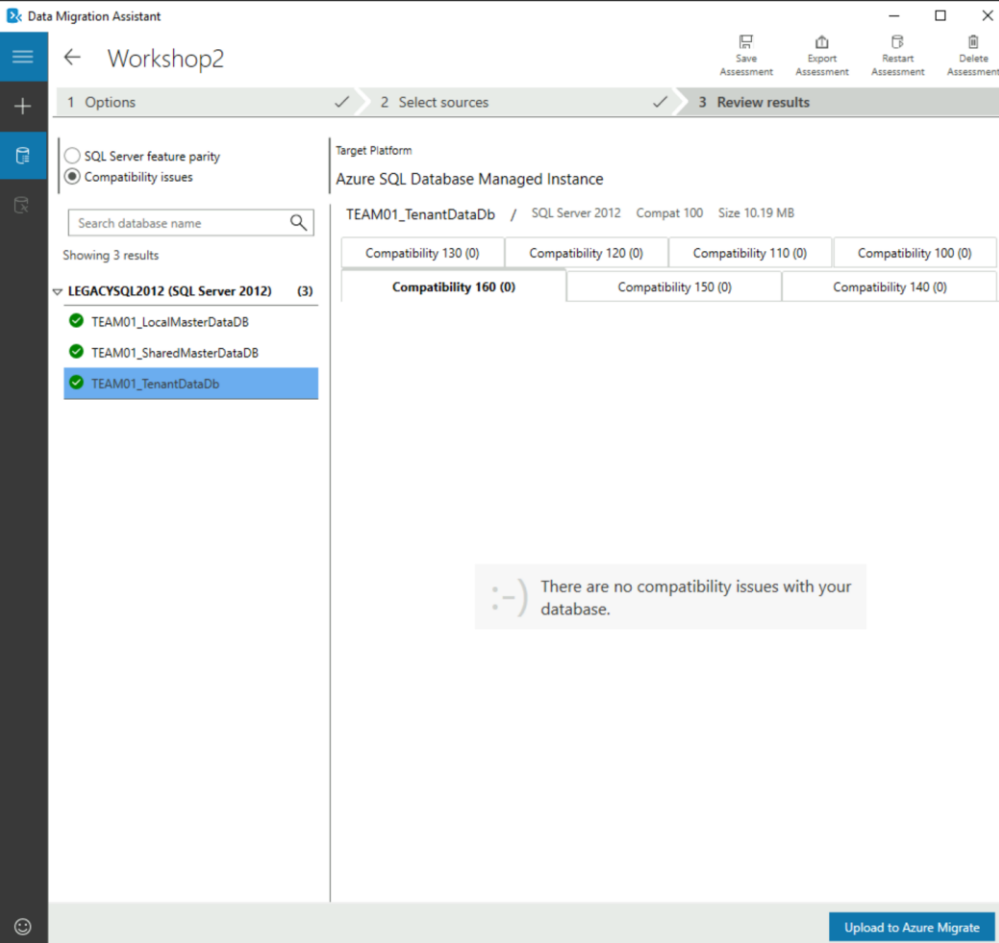
<p>CLR is not supported on Azure SQL DB but is supported by Azure SQL Database Managed Instance (SQLMI).</p>		<p>If you have multiple databases, as with the example screenshot, you need to highlight EACH database to see the compatibility issues.</p>
	<p>Because we need to migrate CLR Stored Procs, we need to repeat the assessment with Azure SQL DB Managed Instance as the target to see if it's compatible</p>	
<p>Once you've reviewed the assessment, click the back arrow near <i>Workshop1</i> Name on the upper left corner to see a list of current DMA projects.</p>		

Azure OpenAI x Databases Hack

<p>You should see a view similar to the screenshot to the right.</p> <p>Select the “+” to create a new assessment project.</p>		
<p>Select/Enter the following details:</p> <p>Project name: <i>Workshop2</i></p> <p>Assessment type: <i>Database Engine</i></p> <p>Source server type: <i>SQL Server</i></p> <p>Target server type: <i>Azure SQL Database Managed Instance</i></p> <p>Click ‘Create’.</p>		<p>Our 2nd assessment project assumes we will be migrating to Azure SQL DB Managed Instance, so the options shown in the screenshot need to be selected.</p>

<p>Select the assessment checks (Report Type) to be made:</p> <ul style="list-style-type: none"> • Check database compatibility • Check feature parity <p>Click 'Next'.</p>		<p>As we saw previously DMA can test for both database compatibility and feature parity compliance against the chosen target.</p> <p>As before we will assess all the databases against all of the tests.</p>
<p>Enter the source/legacy SQL details:</p> <p>Server Name: LEGACYSQL2012</p> <p>Authentication Type: SQL Server Authentication</p> <p>NOTE: Double check that there are no empty spaces in username or password!</p> <p>Username: Demouser</p> <p>Password: Demo@pass1234567</p> <p>Untick "Encrypt connection"</p>		<p>When performing this within your own subscription you will enter the host, authentication and connection types according to your company guidelines and practices.</p> <p><i>Bear in mind that DMA needs to connect to a source SQL Server using an account that belongs to the sysadmin role.</i></p> <p>As this document is produced within a workshop environment Active Directory, Certificates and encryption has not been setup.</p>

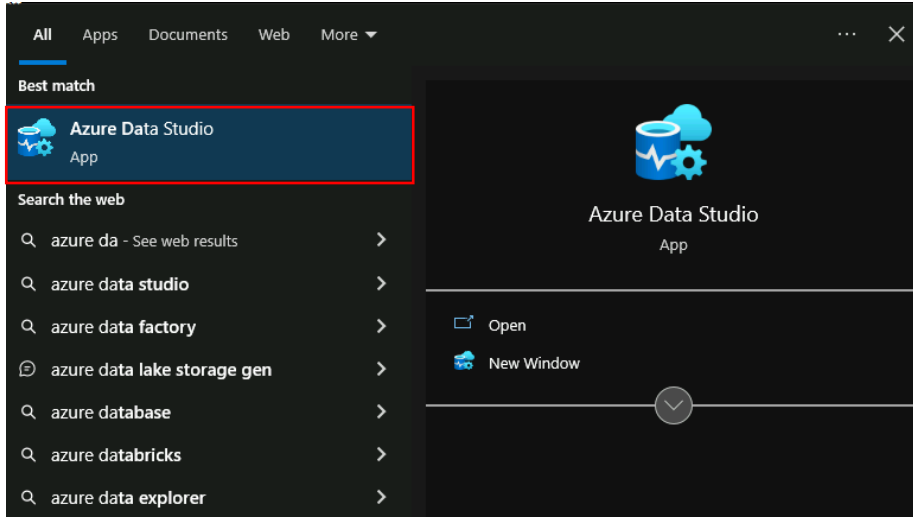
<p>Click 'Connect'.</p>		
<p>Again, select only the 3 database used by your 'Online Transaction Monitor' app. These will have a TEAMXX prefix where XX should be replaced by your team number.</p> <p>TEAMxx_LocalMasterDataDb TEAMxx_SharedMasterDb TEAMxx_TenantDataDb</p> <p>Click 'Add' to add them to the assessment.</p>		<p>DMA will show all databases located on the Source host and display them so you can decide which ones to include in this assessment project.</p> <p>Note that you can assess multiple databases at the same time.</p>
<p>You should now see the screen on the right with the 3 relevant TEAMXX databases listed.</p> <p>Select 'Start Assessment'</p>		<p>Note: DMA allows you to either 'Add' or 'Remove' additional data sources as needed at this point.</p> <p>Also note that DMA has identified what compatibility level each source database is running under.</p>

<p>As before, DMA will now show the results from the assessment as the separate 2 reports.</p> <p>Note the 'SQL Server feature parity' report will either be clean</p> <p>The 'Compatibility Issues' report should be clear for all 3 databases showing that they can be migrated to Azure SQLDB Managed Instance without changes.</p>		<p>Note: Toggle the parity and compatibility Issues radio button (top left) to see how DMA.</p> <p>'SQL Server feature parity' shows what features are not supported in the target data source. Under 'Details' and 'Databases' you will find remedial action that are required and the databases impacted.</p> <p>'Compatibility Issues' shows, over the compatibility tabs, issues that need to be addressed to permit the database(s) to run, in the chosen compatibility level (e.g. 160, 150, 140, 130, 120, 110, 100).</p> <p>If you have multiple databases, as with the example screenshot, you need to highlight EACH database to see the compatibility issues.</p>

	<p>We are now ready to migrate the application databases to Azure SQL Database Managed Instance</p>	
--	--	--

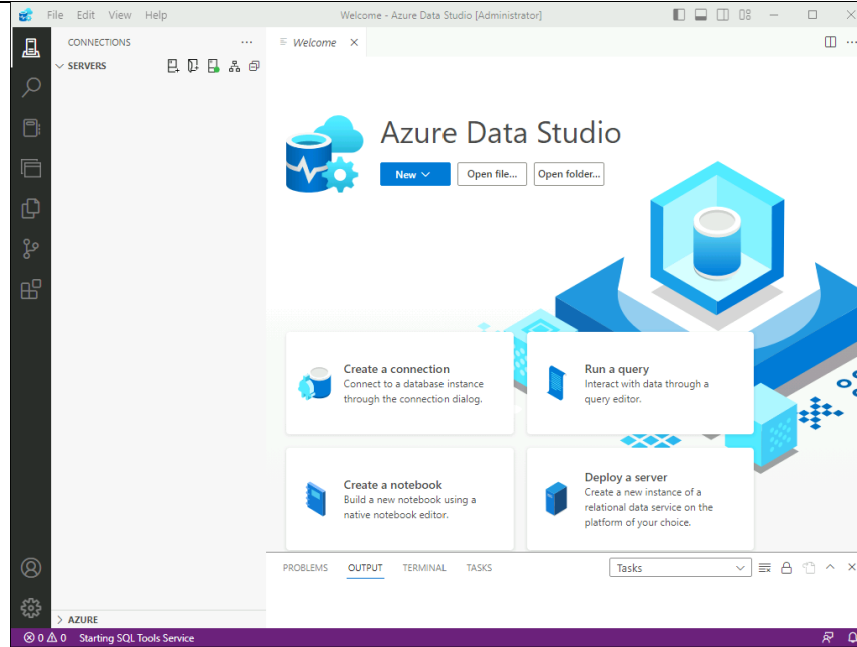
3. Migrate the application databases to Azure SQL Database managed instance using the Azure Data Studio (ADS) with migration extension and identify target Azure SQL SKU

In this section we will use the Azure Data Studio (ADS) to assess the applications database for suitability for migration to Azure Cloud.

Narrative	Screenshot	Notes
<p>We need to determine the suitability of the database(s) for migration to Azure. This includes checking for compatibility and feature support with Azure Database.</p> <p>You should already have an RDP (or Bastion) session open to your teams Win10 Management VM.</p> <p>Now run Azure Data Studio (ADS) from the Start menus or Desktop icon.</p>		<p>Azure Data Studio (ADS) is a free download from Microsoft. It can be used to perform database administration as well as assess a number of database migration & upgrade scenarios not just SQL Server to Azure SQL Database.</p> <p>Download and install Azure Data Studio - Azure Data Studio Microsoft Learn</p>

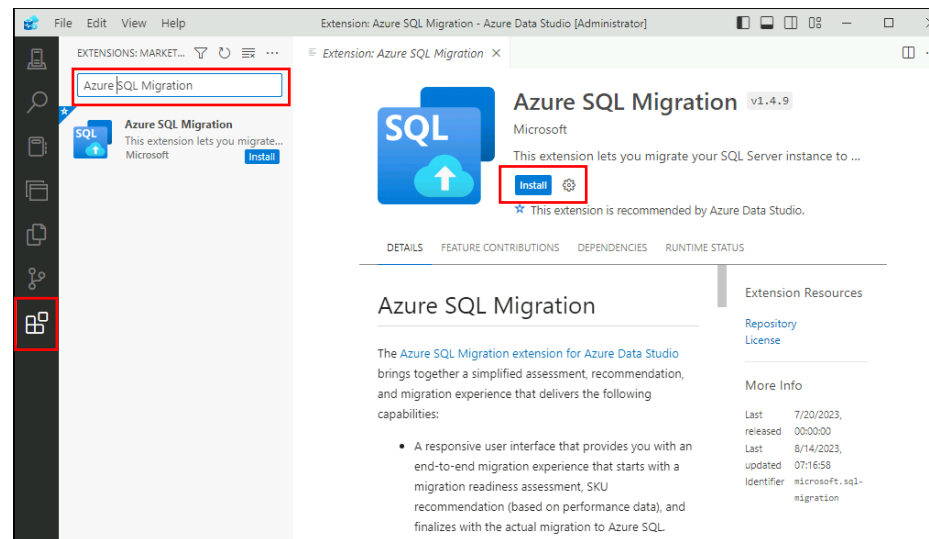
Azure OpenAI x Databases Hack

You should see this screenshot to the right.



Select **“extensions” icon** on the bottom left (or press: CTRL + Shift + X) and search for: **“Azure SQL migration”** in the extension market.
Click **Install**.

Note: If the extension is not compatible with the ADS version installed, upgrade ADS under *Help > Check for Updates*.



See also:

[Azure SQL migration extension for Azure Data Studio - Azure Data Studio | Microsoft Learn](#)

Go back to your [Azure Portal](#) tab, or open your browser and navigate again to:
<https://portal.azure.com>

Login in again with the credential provided at the beginning. Similar to...

User:

sqloihackuserXX@M365x04034558.onmicrosoft.com

Password:

&sqloihack@demo_XX!

Go to the Resource Group **"SQLOIHACK-SHARED"**.

Select the **Azure Storage Account** "sqlhack...".
[sqlhacksaj754o5hum2r36](#)

On the left, click: **Access Keys**

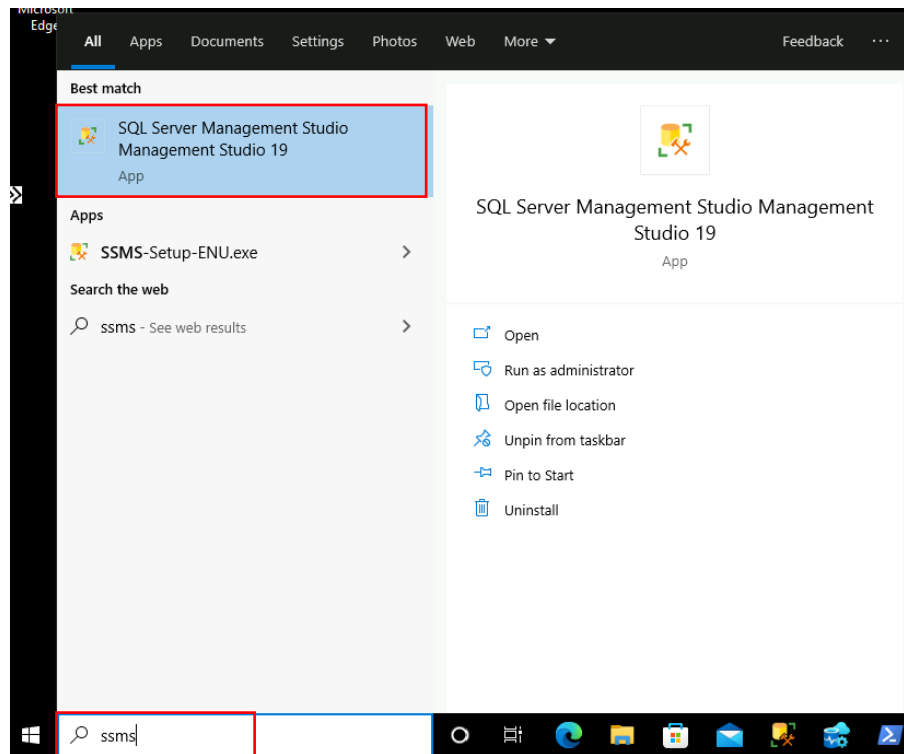
You can note the **Key1** in a notepad to reuse it in following steps.

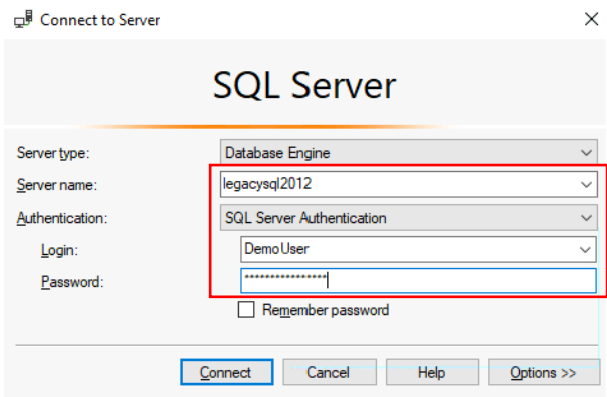
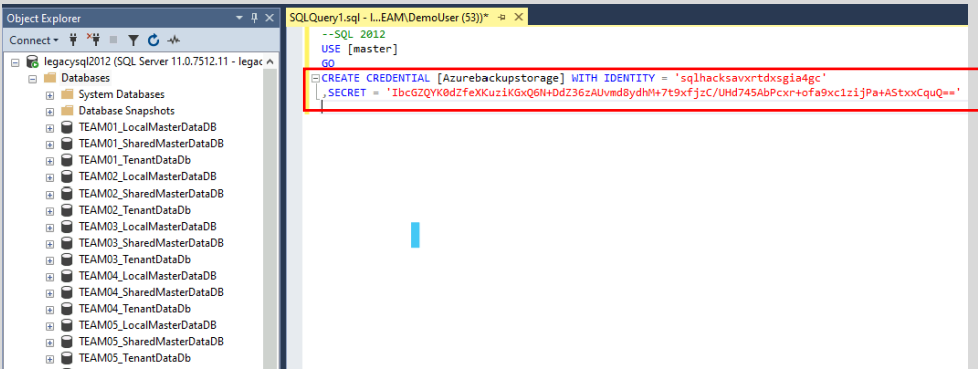
The screenshot shows the 'Access keys' page for the storage account 'sqlhacksavxrt dxsgia4gc'. The left sidebar contains a navigation menu with 'Access keys' highlighted. The main content area shows the storage account name, a 'key1' section with a 'Rotate key' button and a 'Key' field containing a long alphanumeric string, and a 'key2' section with a 'Rotate key' button and a 'Key' field. The 'key1' section is highlighted with a red box.

Azure OpenAI x Databases Hack

Back in the VM, create a database backup in SSMS:

Open **SQL Server Management Studio** (SSMS) on your Team VM by typing it in the start menu bar.



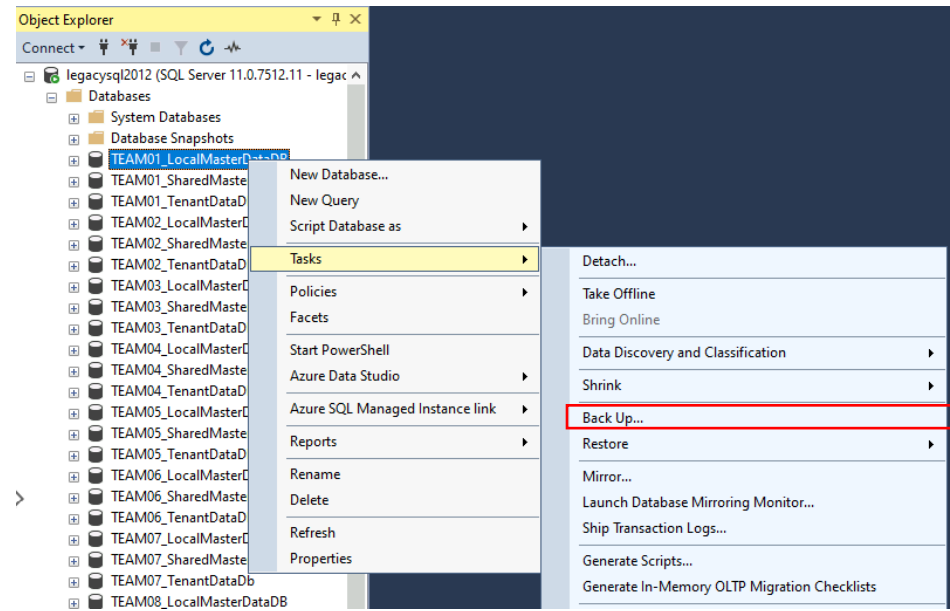
<p>In SSMS, connect to: <i>legacysql2012</i></p> <p>Use the following credentials: User: <i>DemoUser</i> Password: <i>Demo@pass1234567</i></p> <p>NOTE: Double check that there are no empty spaces in username or password!</p>		
<p>DO NOT EXECUTE THIS STEP This is for reference only, as only a single credential is required!</p> <p>In SSMS, open new query and create the credential using the following script:</p> <pre>USE [master] GO CREATE CREDENTIAL [Azurebackupstorage] WITH IDENTITY = '<your storage account name>' ,SECRET = '<your storage account access key>'</pre>		<p>This should be only done only by the trainer.</p>

Backup your team databases:

Select **your team's 3 databases** and create a **full back** to URL for each database:

- 1) **TEAMXX_TenantDataDB**
- 2) **TEAMXX_LocalMasterDataDB**
- 3) **TEAMXX_SharedMasterDataDB**

You can do so by right-click on the first database, select tasks > Back Up...



This is the wizard experience in SSMS, you can also take backups using T-SQL scripts. There are some samples below, for this.

Backup database:

- 1) Select **Backup to URL**
- 2) Select the **credential** *“MigrationContainerCred”*
- 3) Make sure you enter the **Azure container name** as follows:
migration/team<XX>_<database>

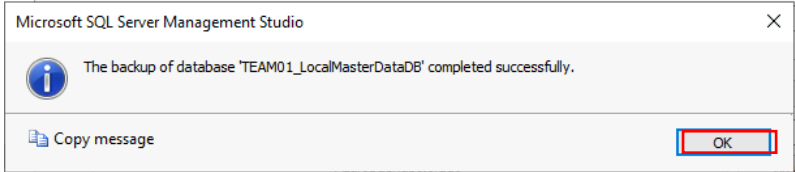
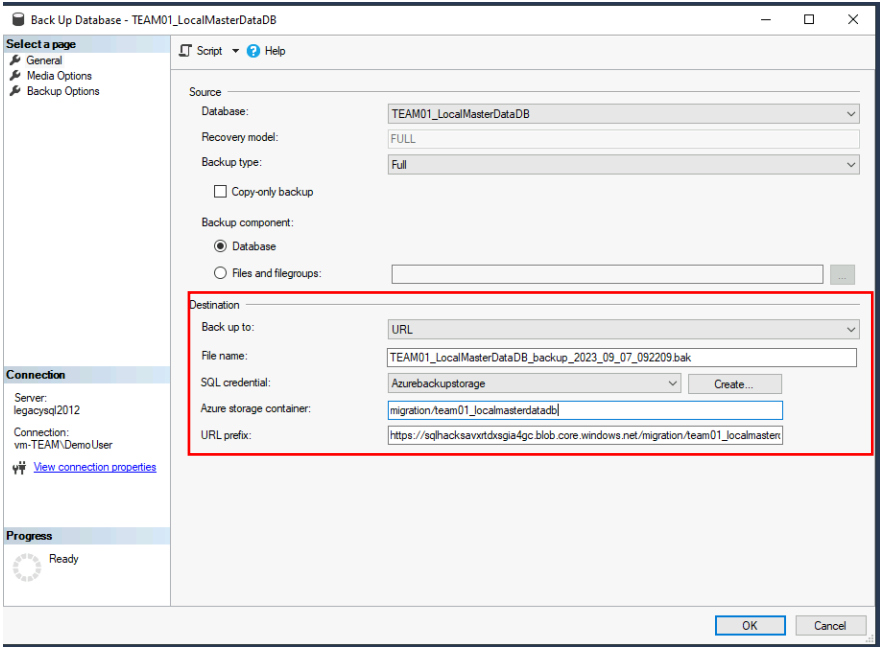
e.g. for team01
migration/team01_localmasterdatadb

Repeat this process for the remaining 2 databases.

Make sure to name the Azure container name in step 3 according to the databases.

- TEAMXX_LocalMasterDataDB
- TEAMXX_SharedMasterDataDB
- TEAMXX_TenantDataDB

Use SSMS like above or use TSQL commands in the right-hand side.



You can also directly use TSQL to BackUp your Databases:

BACKUP DATABASE [TEAM01_LocalMasterDataDB] TO URL = N'https://sqlhacksavxrtdxsgia4gc.blob.core.windows.net/migration/team01_localmasterdatadb/TEAM01_LocalMasterDataDB_backup_2023_09_07_092209.bak' WITH CREDENTIAL = N'Azurebackupstorage', NOFORMAT, NOINIT, NAME = N'TEAM01_LocalMasterDataDB-Full Database Backup', NOSKIP, NOREWIND, NOUNLOAD, STATS = 10

GO

BACKUP DATABASE [TEAM01_SharedMasterDataDB] TO URL = N'https://sqlhacksavxrtdxsgia4gc.blob.core.windows.net/migration/team01_shareddatadb/TEAM01_SharedMasterDataDB_backup_2023_09_07_092209.bak' WITH CREDENTIAL = N'Azurebackupstorage', NOFORMAT, NOINIT, NAME = N'TEAM01_SharedMasterDataDB-Full Database Backup', NOSKIP, NOREWIND, NOUNLOAD, STATS = 10

GO

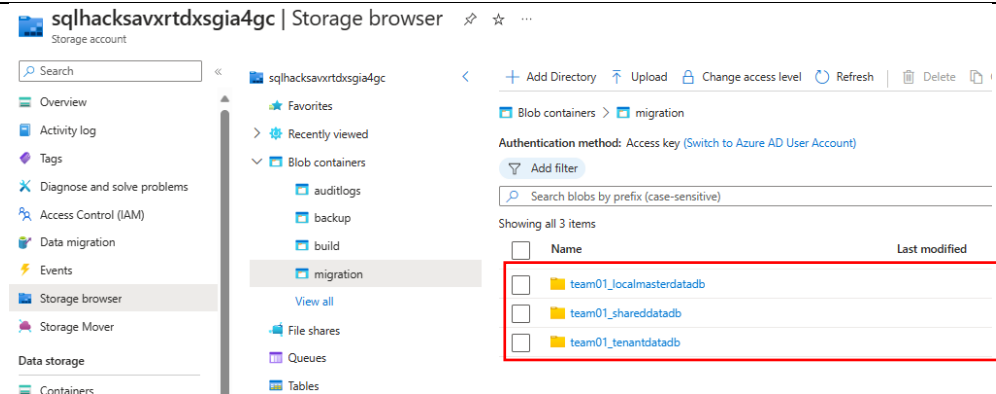
BACKUP DATABASE [TEAM01_TenantDataDB] TO URL = N'https://sqlhacksavxrtdxsgia4gc.blob.core.windows.net/migration/team01_tenantdatadb/TEAM01_TenantDataDB_backup_2023_09_07_092209.bak' WITH CREDENTIAL = N'Azurebackupstorage', NOFORMAT, NOINIT, NAME = N'TEAM01_TenantDataDB-Full Database Backup', NOSKIP, NOREWIND, NOUNLOAD, STATS = 10

GO

Switch to Azure portal on your web browser.		
--	--	--

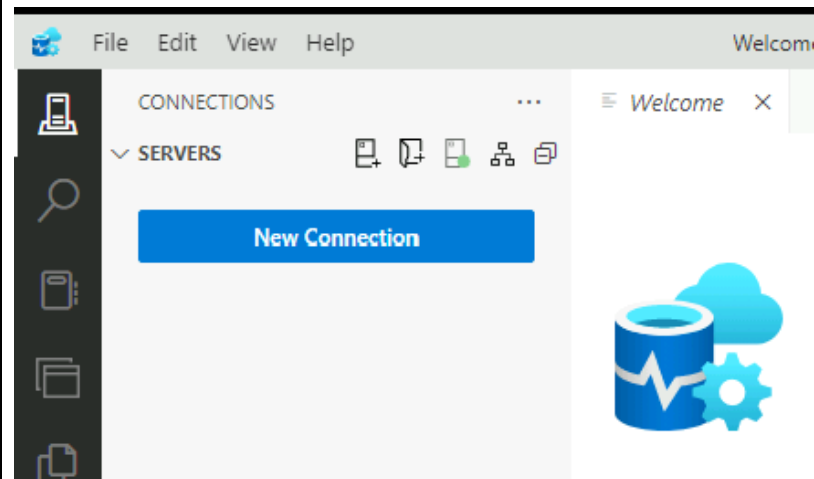
Azure OpenAI x Databases Hack

Review and check for existence of the **full backup** in the Azure Storage account in each folder of the 3 databases.



Switch to the **Azure Data Studio** on your Team VM.

Connect to legacy SQL Server 2012 using “**New Connection**” in the upper left corner.



Enter server name and credentials.

Server:

legacysql2012

User:

DemoUser

Password:

Demo@pass1234567

Trust server certificate:

True

And click **Connect**.

NOTE: Double check that there are no empty spaces in username or password!

Connection Details

Connection type: Microsoft SQL Server

Input type: ☒ Parameters ☐ Connection String

Server*: legacysql2012

Authentication type: SQL Login

User name*: DemoUser

Password: *****

☐ Remember password

Database: <Default>

Encrypt: Mandatory (True)

Trust server certificate: True

Server group: <Default>

Name (optional):

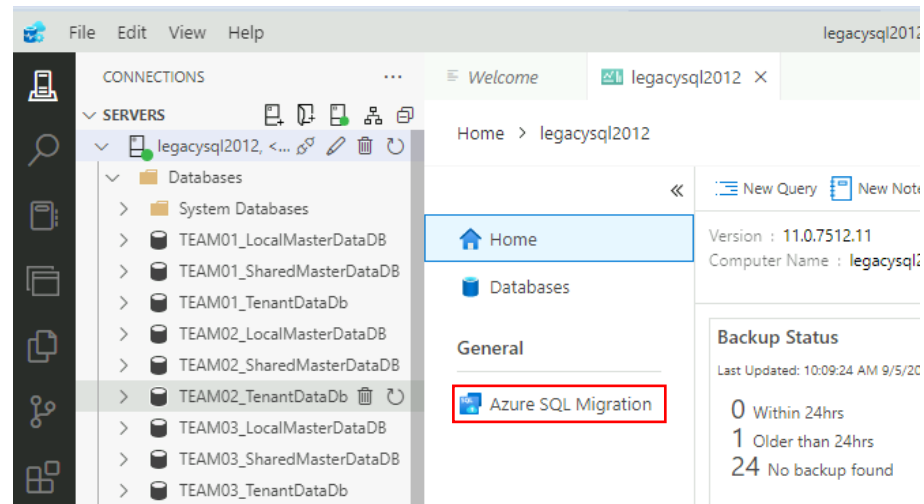
Advanced...

Connect Cancel

Azure OpenAI x Databases Hack

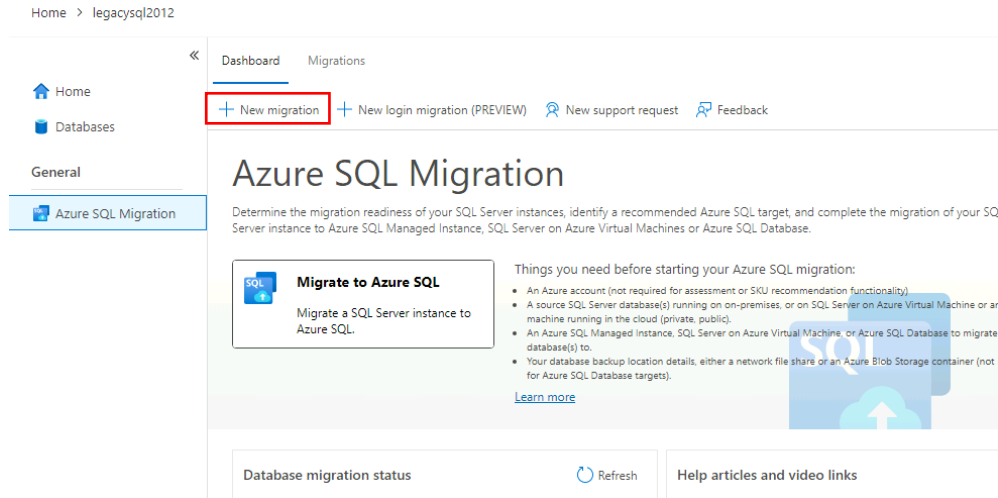
(Optional if you don't see the view yet)
Right Click on the SQL Server instance **legacysql2012** on the left-hand side and select **"Manage"**.

Then click **"Azure SQL Migration"** under General.



Azure OpenAI x Databases Hack

In the tab **Azure SQL migration**, now choose + “**New migration**”.



The screenshot displays the Azure SQL Migration dashboard. The breadcrumb path is 'Home > legacysql2012'. The left sidebar contains a 'Home' link, a 'Databases' link, and a 'General' section with an 'Azure SQL Migration' link. The main content area has a 'Dashboard' tab and a 'Migrations' tab. In the 'Migrations' tab, there is a '+ New migration' button highlighted with a red box, followed by '+ New login migration (PREVIEW)', 'New support request', and 'Feedback'. Below this, the 'Azure SQL Migration' section is titled, with a description: 'Determine the migration readiness of your SQL Server instances, identify a recommended Azure SQL target, and complete the migration of your SQL Server instance to Azure SQL Managed Instance, SQL Server on Azure Virtual Machines or Azure SQL Database.' A card titled 'Migrate to Azure SQL' contains the text 'Migrate a SQL Server instance to Azure SQL.' To the right, a list of requirements is provided: 'Things you need before starting your Azure SQL migration:'. The requirements are: 'An Azure account (not required for assessment or SKU recommendation functionality)', 'A source SQL Server database(s) running on on-premises, or on SQL Server on Azure Virtual Machine or on machine running in the cloud (private, public)', 'An Azure SQL Managed Instance, SQL Server on Azure Virtual Machine, or Azure SQL Database to migrate database(s) to', and 'Your database backup location details, either a network file share or an Azure Blob Storage container (not for Azure SQL Database targets)'. A 'Learn more' link is provided. At the bottom, there is a 'Database migration status' section with a 'Refresh' button and a 'Help articles and video links' section.


Home > legacysql2012

Dashboard Migrations

+ New migration + New login migration (PREVIEW) New support request Feedback

Azure SQL Migration

Determine the migration readiness of your SQL Server instances, identify a recommended Azure SQL target, and complete the migration of your SQL Server instance to Azure SQL Managed Instance, SQL Server on Azure Virtual Machines or Azure SQL Database.

**Migrate to Azure SQL**
Migrate a SQL Server instance to Azure SQL.

Things you need before starting your Azure SQL migration:

- An Azure account (not required for assessment or SKU recommendation functionality).
- A source SQL Server database(s) running on on-premises, or on SQL Server on Azure Virtual Machine or on machine running in the cloud (private, public).
- An Azure SQL Managed Instance, SQL Server on Azure Virtual Machine, or Azure SQL Database to migrate database(s) to.
- Your database backup location details, either a network file share or an Azure Blob Storage container (not for Azure SQL Database targets).

[Learn more](#)

Database migration status [Refresh](#) [Help articles and video links](#)

Select **your 3 team databases**:

- 1) TEAMXX_TenantDataDB
- 2) TEAMXX_LocalMasterDataDB
- 3) TEAMXX_SharedMasterDataDB

And click **Next**.

Migrate 'legacysql2012' to Azure SQL

Step 1: Databases for assessment

Select the databases that you want to assess for migration to Azure SQL.

Search

3/49 databases selected

<input type="checkbox"/>	Database	Status	Size (MB)
<input checked="" type="checkbox"/>	TEAM01_LocalMasterDataDB	ONLINE	3
<input checked="" type="checkbox"/>	TEAM01_SharedMasterDataDB	ONLINE	5
<input checked="" type="checkbox"/>	TEAM01_TenantDataDb	ONLINE	10
<input type="checkbox"/>	TEAM02_LocalMasterDataDB	ONLINE	3
<input type="checkbox"/>	TEAM02_SharedMasterDataDB	ONLINE	3
<input type="checkbox"/>	TEAM02_TenantDataDb	ONLINE	10
<input type="checkbox"/>	TEAM03_LocalMasterDataDB	ONLINE	3
<input type="checkbox"/>	TEAM03_SharedMasterDataDB	ONLINE	3
<input type="checkbox"/>	TEAM03_TenantDataDb	ONLINE	10
<input type="checkbox"/>	TEAM04_LocalMasterDataDB	ONLINE	3
<input type="checkbox"/>	TEAM04_SharedMasterDataDB	ONLINE	3
<input type="checkbox"/>	TEAM04_TenantDataDb	ONLINE	10
<input type="checkbox"/>	TEAM05_LocalMasterDataDB	ONLINE	3
<input type="checkbox"/>	TEAM05_SharedMasterDataDB	ONLINE	3
<input type="checkbox"/>	TEAM05_TenantDataDb	ONLINE	10
<input type="checkbox"/>	TEAM06_LocalMasterDataDB	ONLINE	3
<input type="checkbox"/>	TEAM06_SharedMasterDataDB	ONLINE	3

Assess extended event sessions

Next

Cancel

Run **assessment** and receive recommendations.

Click **Next**.

Check provided recommendations.

Migrate 'legacysql2012' to Azure SQL

Step 2: Assessment summary and SKU recommendations

Refresh Assessment | Refresh SKU | Start data collection | Stop data collection | Import performance data | Recommendation parameters

✓ We have completed the assessment of your SQL Server instance 'legacysql2012'.

View assessment summary and SKU recommendations for Azure SQL targets

Target	Ready	Needs review	Not ready	Blockers	Warnings
Azure SQL Database (PaaS)	2	0	1	2	3
Azure SQL Managed Instance (PaaS)	3	0	0	0	1
SQL Server on Azure Virtual Machine (IaaS)	3	0	0	0	0

RECOMMENDED CONFIGURATION ⓘ

Azure recommendation is not available. Click 'Start data collection' button above to get started.

Select the **target type** as “**Azure SQL Managed Instance**”.

Click **Next**.

Migrate 'legacysql2012' to Azure SQL

Step 3: Target platform & assessment results

Choose target platform, view assessment results and select database(s) for migration.

Select target type: Azure SQL Managed Instance

Recommended configuration

Database(s) assessed

Ready for migration

--

3

3

Search

Assessment findings

Details

Instance

Findings

Summary

Findings

Details

legacysql2012

1

Trace flags not supported in...

Assessment summary

Databases (3/3)

Database

Findings

Migration readiness of assessed databases in the server instance

TEAM01_TenantDataDb

0

Assessed databases: 3

TEAM01_LocalMasterDataDB

0

Not ready (0)

TEAM01_SharedMasterDataDB

0

Ready with warnings (0)

Ready (3)

Server instance assessment findings summary

Total findings: 1

For this, you need to **add your account** that you use to login to Azure Portal:

Similar to...

User:

sqloaihackuserXX@M365x04034558.onmicrosoft.com

Password:

&sqloaihack@demo_XX!

CONNECTIONS

Migrate 'legacysql2012' to Azure SQL

Step 3: Azure SQL target

Select an Azure account and your target Azure SQL Managed instance.

Azure account*

Add a linked account and then try again.

[Link account](#)

Subscription*

Select a subscription

Location*

Select a location

Resource group*

Select a resource group

Azure SQL Managed Instance*

Select a target server.

Linked accounts

There is no linked account. Please add an account.

[Add an account](#)

Adding account...

Close

Previous Next Cancel

Select the **Azure subscription**, the **Location**, **Resource Group** and **Azure SQL MI FDQN Name** which are automatically provided from your Account.

And Click **Next**.

CONNECTIONS

Migrate 'legacysql2012' to Azure SQL

Step 4: Azure SQL target

Select an Azure account and your target Azure SQL Managed Instance (PaaS).

Azure account*

sqloaihackuser02 - sqloaihackuser02@M365x04034558.onmicrosoft.com

[Link account](#)

Subscription*

Azure Pass - Sponsorship - f6eb7b8d-1599-4cab-a223-0245e015a540

Location*

East US

Resource group*

sqloaihack-shared

Azure SQL Managed Instance*

sqihackmi-j754o5hum2r36

Linked accounts

There is no linked account. Please add an account.

[Add an account](#)

Adding account...

Close

Previous Next Cancel

Select “**offline migration**”.

Select the existing **Azure Database Migration Service**:

sqlhack-dmsV2

Migrate 'legacysql2012' to Azure SQL

Step 4: Azure Database Migration Service

☐ Online migration

Application downtime is limited to cutover at the end of migration.

☒ Offline migration

Application downtime will start when the migration starts.

Select the location of the database backups to use during migration.

☒ My database backups are in an Azure Storage Blob Container

☐ My database backups are on a network share

Azure Database Migration Service orchestrates database migration activities and tracks their progress. You can select an existing Database Migration Service if you have created one previously, or create a new one below.

Subscription

ME-MngEnvMCAP072286-csukalla-1 - 5d0d9ca7-87be-4108-b978-a75cf10f1b95

Location

West Europe

Resource group *

sqlhack-shared

Azure Database Migration Service *

sqlhack-dmsV2

[Create new](#)

You can also do an Online Migration for mission critical workloads using DMS. There are additional steps that you should take for this. Please use the information in the following tutorial for Online Migration:

[Tutorial: Migrate SQL Server to Azure SQL Managed Instance online by using Azure Data Studio - Azure Database Migration Service | Microsoft Learn](#)

(You can also create a new Database Migration Service within minutes of you do the exercise in your own subscription. For this you can click on “create new in Step 6”)

In the data source configuration, change the column “**blob container**” to “**migration**” for all 3 databases.

Then select the “**last full backup file**” from **your team** for each of the 3 databases.

E.g. as shown in the screenshot, the Team01 specific database files were selected. Make sure to check that for the TenantDataDB, the respective tenantdatadb backup file is selected. Same for the LocalMasterDataDB and SharedMasterDataDB.

Check the example in screenshot on the right.

Click **Next**.

Migrate 'legacysql2012' to Azure SQL

Step 5: Data source configuration

Azure Storage Blob Container details
Provide the Azure Storage Blob Container that contains the backups.

Subscription
ME-MngEnvMCAPO72286-csukalla-1

Location
West Europe

Enter target database name and select resource group, storage account and container for the selected source databases.
All fields are required.

! When uploading database backups to your blob container, ensure that backup files from different databases are stored in separate folders. Only the root of the container and folders at most one level deep are supported.

Source database name	Target database name	Resource group	Storage account	Blob container	Last backup file
TEAM01_TenantDataDb	TEAM01_TenantDataDb	sqlhack-shared	sqlhacksavvrtdxsgia4gc	migration	team01_tenantdatad...
TEAM01_LocalMasterDataDB	TEAM01_LocalMasterDa...	sqlhack-shared	sqlhacksavvrtdxsgia4gc	migration	team01_localmasterd...
TEAM01_SharedMasterDataDB	TEAM01_SharedMasterD...	sqlhack-shared	sqlhacksavvrtdxsgia4gc	migration	team01_shreddatad...

Review summary and **start migration.**

Migrate 'legacysql2012' to Azure SQL

Step 6: Summary

Source databases

Databases for migration [2](#)

Azure SQL target

Azure account System Administrator -
admin@MngEnvMCAP072286.onmicrosoft.com

Azure SQL target Azure SQL Managed Instance

Subscription ME-MngEnvMCAP072286-csukalla-1

Location West Europe

Resource group sqlhack-shared

Azure SQL Managed Instance sqlhackmi-vrtdxsgia4gc

Migration mode

Mode Offline migration

Data source configuration

Type Blob container

Azure storage subscription ME-MngEnvMCAP072286-csukalla-1

Azure Database Migration Service

Subscription ME-MngEnvMCAP072286-csukalla-1

Location West Europe

Resource group sqlhack-shared

Azure Database Migration Service sqlhack-dmsV2

Previous

Start migration

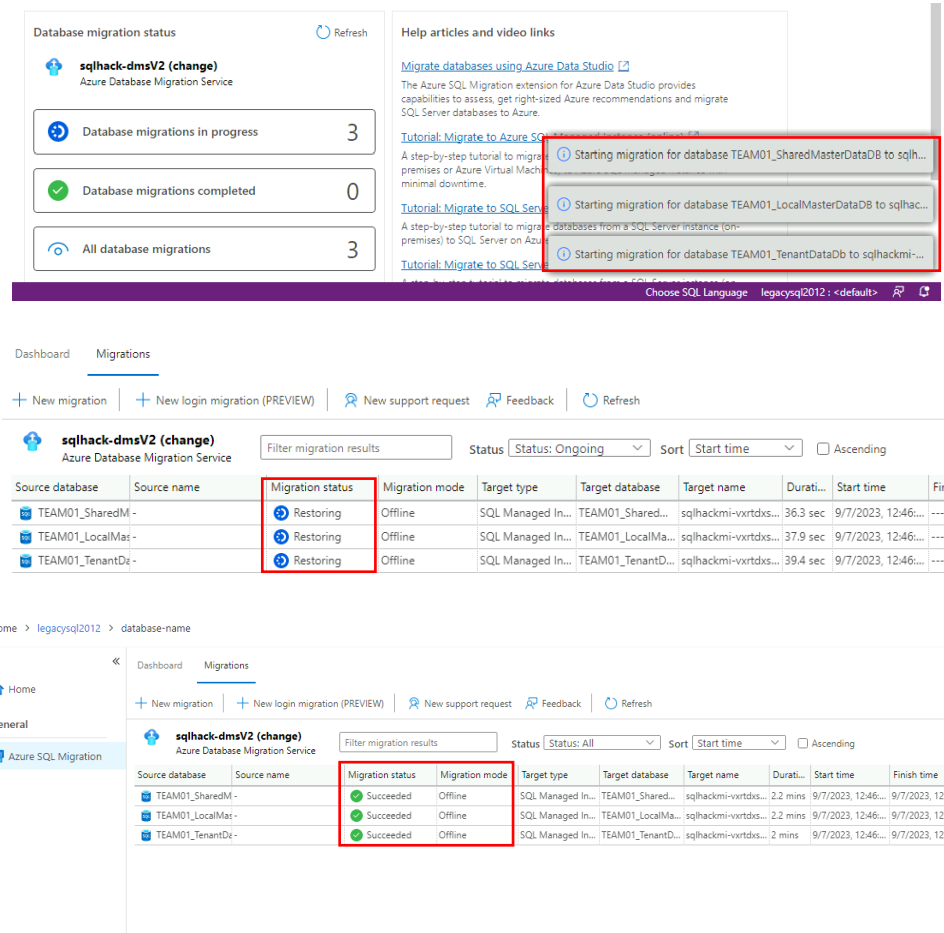
Cancel

The migration may take some time.

Review progress in **Azure Data Studio** under **Azure SQL Migration**.

Click on **Refresh** from time to time to check the latest status of the migration until it succeeds.

To check the progress in real time, click on **All Database Migrations** button and monitor the status of the migration process.



Database migration status

sqlhack-dmsV2 (change)
Azure Database Migration Service

Database migrations in progress: 3

Database migrations completed: 0

All database migrations: 3

Help articles and video links

Migrate databases using Azure Data Studio

Tutorial: Migrate to Azure SQL

Tutorial: Migrate to SQL Server

Starting migration for database TEAM01_SharedMasterDataDB to sqlhackmi-vxrtidxs...

Starting migration for database TEAM01_LocalMasterDataDB to sqlhackmi-vxrtidxs...

Starting migration for database TEAM01_TenantDataDB to sqlhackmi-vxrtidxs...

Choose SQL Language: legacysql2012; <default>

Dashboard | Migrations

+ New migration | + New login migration (PREVIEW) | New support request | Feedback | Refresh

sqlhack-dmsV2 (change)
Azure Database Migration Service

Filter migration results

Status: Ongoing | Sort: Start time | Ascending

Source database	Source name	Migration status	Migration mode	Target type	Target database	Target name	Duration	Start time	Finish time
TEAM01_SharedM-		Restoring	Offline	SQL Managed In...	TEAM01_Shared...	sqlhackmi-vxrtidxs...	36.3 sec	9/7/2023, 12:46:...	---
TEAM01_LocalMas-		Restoring	Offline	SQL Managed In...	TEAM01_LocalMa...	sqlhackmi-vxrtidxs...	37.9 sec	9/7/2023, 12:46:...	---
TEAM01_TenantD-		Restoring	Offline	SQL Managed In...	TEAM01_TenantD...	sqlhackmi-vxrtidxs...	39.4 sec	9/7/2023, 12:46:...	---

Home > legacysql2012 > database-name

Dashboard | Migrations

+ New migration | + New login migration (PREVIEW) | New support request | Feedback | Refresh

sqlhack-dmsV2 (change)
Azure Database Migration Service

Filter migration results

Status: All | Sort: Start time | Ascending

Source database	Source name	Migration status	Migration mode	Target type	Target database	Target name	Duration	Start time	Finish time
TEAM01_SharedM-		Succeeded	Offline	SQL Managed In...	TEAM01_Shared...	sqlhackmi-vxrtidxs...	2.2 mins	9/7/2023, 12:46:...	9/7/2023, 12:49:...
TEAM01_LocalMas-		Succeeded	Offline	SQL Managed In...	TEAM01_LocalMa...	sqlhackmi-vxrtidxs...	2.2 mins	9/7/2023, 12:46:...	9/7/2023, 12:49:...
TEAM01_TenantD-		Succeeded	Offline	SQL Managed In...	TEAM01_TenantD...	sqlhackmi-vxrtidxs...	2 mins	9/7/2023, 12:46:...	9/7/2023, 12:48:...

4. Confirm application databases have been migrated to Azure SQL Managed Instance

In SSMS connect to the source server: **LEGACYSQL2012** and the target server: **sqlolaihackmi-xxxxx.xxxxxxx.database.windows.net** and review the migrated databases.

The target server should be: **sqlhackmi-j754o5hum2r36.7a59bf01d694.database.windows.net**

For connecting, use the credentials:
User: **DemoUser**
Password: **Demo@pass1234567**

NOTE: Double check that there are no empty spaces in username or password!

