SQL Modernisation Hack

Database Migration Lab Step-by-step

(Using Azure SQL MI Link)

V3.0

Contents

[Migration architecture and Azure components A white background with black dots

Description automatically generated 2](#_Toc145418697)

[Generic Migration Content 3](#_Toc145418698)

[1. Get the SQL Managed Instance FQDN 4](#_Toc145418699)

[2. Assess the application databases for Azure SQL Database suitability using the Database Migration Assistant (DMA) 5](#_Toc145418700)

[3. Identify target Azure SQL SKU with Data Migration Assistant(DMA) 16](#_Toc145418701)

[4. Use Link feature for Azure SQL Managed Instance to migrate a database 19](#_Toc145418702)

# Migration architecture and Azure components A white background with black dots Description automatically generated

A screenshot of a computer

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SQLHACK-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** |
| *Notes for outside of the workshop:*  *Familiarise yourself with Microsoft migration tools and the Azure Database Migration Guide* | Azure Database Migration Guide:  [https://www.microsoft.com/en-us/download/default.aspx](https://azure.microsoft.com/en-gb/services/database-migration/)  DMA & download link:  <https://docs.microsoft.com/en-us/sql/dma/dma-overview?view=sql-server-ver15>  Microsoft Migration Portal:  [https://datamigration.microsoft.com/](https://www.microsoft.com/en-us/download/default.aspx)  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> |

# 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** |
| On your Win10 VM open Edge browser and got to:  [**HTTPS://portal.azure.com**](HTTPS://portal.azure.com)  **Username and Password:**  *(see your Teams Group)*    In the Azure portal, open the **SQLHACK-SHARED** **Resource Group** and locate the **SQL managed instance** and open it.  **Note the Host Name (FQDN)**  sqlhackmi-xxxxx.xxxxxxx.database.windows.net  All other **details from the “DB Migration Lab and Parameters.pdf”** |  |  |

# 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** |
| 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.  You should already have a remote (Bastion) session open to your teams Win10 Management VM**,** if sorun DMA from the Start menus or Desktop icon. | A screen shot of a computer  Description automatically generated | 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. |
| You should see this screenshot to the right.  Select the **“+”** to create a new **assessment** project |  |  |
| Select/Enter the following details:  **Project name:**  **Workshop1**  **Assessment type:**  **Database Engine**  **Source server type:**  **SQL Server**  **Target server type:**  **Azure SQL Database**  Click ‘**Create’** |  | Our first project assessment assumes we will be migrating to Azure SQL DB, so the options shown in the screenshot need to be selected. |
| Select the assessment checks (Report Type) to be made:  **Check database compatibility**  **Check feature parity**  Click ‘**Next’** |  | DMA can test for both database compatibility and feature parity compliance against the Azure target.  As this is the initial evaluation, we are assessing a database(s) we will perform all of these tests. |
| Enter the source/legacy SQL details:  **Server Name:**  **LEGACYSQL2012**  **Authentication Type:**  **SQL Server Authentication**  **Username:**  **Demouser**  **Password:**  **Demo@pass1234567**  **Untick “Encrypt connection”**  Click ‘**Connect’**  ***If you get an error logging in check that the Win10 keyboard language*** |  | When performing this within your own subscription you will enter the host, authentication and connection types according to your company guidelines and practices.  *Bear in mind that DMA needs to connect to a source SQL Server using an account that belongs to the* ***sysadmin*** *role.*  As this document is produced within a workshop environment Active Directory, Certificates and encryption has not been setup. |
| 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.  **TEAMxx\_LocalMasterDataDb**  **TEAMxx\_SharedMasterDb**  **TEAMxx\_TenantDataDb**  Click ‘**Add’** to add them to the assessment. |  | 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.  Note that you can assess multiple databases at the same time. |
| You should now see the screen on the right with the relevant TEAMxx databases listed.  Select ‘**Start Assessment’** |  | Note: DMA allows you to either ‘Add’ or ‘Remove’ additional data sources as needed at this point.  Also note that DMA provides some high-level metadata about the databases including their compatibility level the total size of each database.  [Using Data Migration Assistant to assess an application’s data access layer](https://techcommunity.microsoft.com/t5/microsoft-data-migration/using-data-migration-assistant-to-assess-an-application-s-data/ba-p/990430) |
| DMA will now show the results of the assessment using 2 separate reports:  ‘**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**).  ‘**Compatibility Issues**’ which is a database level report detailing individual objects that have compatibility issues.  Select ‘**TEAMxx\_TenantDataDb**’ Note the ‘Migration blockers’ and “Breaking Changes” including CLR which the database uses.  CLR is not supported on Azure SQL DB but is supported by Azure SQL Database Managed Instance (SQLMI). |  | **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).  If you have multiple databases, as with the example screenshot, you need to highlight EACH database to see the compatibility issues. |
|  |  |  |
|  | **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** |  |
| Once you’ve reviewed the assessment click the back arrow to see a list of current DMA projects.  You should see this screenshot to the right.  Select the **“+”** to create a new **assessment** project. |  |  |
| Select/Enter the following details:  **Project name:**  **Workshop2**  **Assessment type:**  **Database Engine**  **Source server type:**  **SQL Server**  **Target server type:**  **Azure SQL Database Managed Instance**  Click ‘**Create’** |  | 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. |
| Select the assessment checks (Report Type) to be made:  **Check database compatibility**  **Check feature parity**  Click ‘**Next’** |  | As we saw previously DMA can test for both database compatibility and feature parity compliance against the chosen target.  As before we will assess all the databases against all of the tests. |
| Enter the source/legacy SQL details:  **Server Name:**  **LEGACYSQL2012**  **Authentication Type:**  **SQL Server Authentication**  **Username:**  **Demouser**  **Password:**  **Demo@pass1234567**  **Untick “Encrypt connection”**  Click ‘**Connect’** |  | When performing this within your own subscription you will enter the host, authentication and connection types according to your company guidelines and practices.  *Bear in mind that DMA needs to connect to a source SQL Server using an account that belongs to the sysadmin role.*  As this document is produced within a workshop environment Active Directory, Certificates and encryption has not been setup. |
| 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.  **TEAMxx\_LocalMasterDataDb**  **TEAMxx\_SharedMasterDb**  **TEAMxx\_TenantDataDb**  Click ‘**Add’** to add them to the assessment. |  | 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.  Note that you can assess multiple databases a t the same time. |
| You should now see the screen on the right with the relevant TEAMXX databases listed.  Select ‘**Start Assessment’** |  | Note: DMA allows you to either ‘Add’ or ‘Remove’ additional data sources as needed at this point.  Also note that DMA has identified what compatibility level each source database is running under. |
|  |  |  |
| As before DMA will now show the results from the assessment as the separate 2 reports.  Note the ‘**SQL Server feature parity**’ report will either be clean  The ‘**Compatibility Issues**’ report should be clear for all 3 databases showing that they can be migrated to Azure SQLDB Managed Instance without changes. |  | Note: Toggle the parity and compatibility Issues radio button (top left) to see how DMA.  ‘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.  ‘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).  If you have multiple databases, as with the example screenshot, you need to highlight **EACH** database to see the compatibility issues. |
|  | **We are now ready to migrate the application databases to Azure SQL Database Managed Instance** |  |

# Identify target Azure SQL SKU with Data Migration Assistant(DMA)

For an overview on DMA Sizing recommendation tool, consult the following link:

[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)

The 3 stages:

1. Collect performance data
2. Get SKU recommendation
3. Display the generated HTML SKU recommendation report

|  |  |  |
| --- | --- | --- |
| **Narrative** | **Screenshot** | **Notes** |
| On your Win10 VM open Command Prompt  Select the following folder:  **C:\Program Files\Microsoft Data Migration Assistant\SqlAssessmentConsole**  All the command needed after are in the following file: **DMA\_Sizing\_CommandLines.txt** |  |  |
| **1.Collect performance Data**  Execute the following command to collect performance data:  **SqlAssessment.exe PerfDataCollection --sqlConnectionStrings "Data Source=legacySQL2012;Initial Catalog=master;Integrated Security=True;" --outputFolder C:\Output**  At least 20 iterations are needed (10 minutes) |  | For more details on how to collect data:  [**DMA Collect performance Data**](https://docs.microsoft.com/en-us/sql/dma/dma-sku-recommend-sql-db?view=sql-server-ver15#collect-performance-data) |
| **2.** **Get SKU recommendation**  Execute the following command to generate the recommendation:  **SqlAssessment.exe GetSkuRecommendation --OutputFolder C:\Output --targetPlatForm AzureSQLManagedInstance**  Or with a more representative performance dataset:  **SqlAssessment.exe GetSkuRecommendation --OutputFolder C:\\_SQLHACK\_\LABS\01-Data\_Migration\DMA\_Sizing**  **--targetPlatForm AzureSQLManagedInstance** |  | For more details on how to get the SQL SKU recommendation:  [Use the console application to get SKU recommendations](https://docs.microsoft.com/en-us/sql/dma/dma-sku-recommend-sql-db?view=sql-server-ver15#use-the-console-application-to-get-sku-recommendations) |
| **3. Display the generated HTML SKU recommendation report**  From the **OutputFolder**, **open** the **SkuRecommendationReport.html**  You can have a look to the detailed information inside the file  **SkuRecommendationReport.json** |  |  |

# Use Link feature for Azure SQL Managed Instance to migrate a database

Until now, we have evaluated LEGACYSQL2012 for its potential to migrate to the cloud and collected some recommendations for the target database and SKU (SQL Managed Instance as the deployment option, along with it’s tier, size, etc.)

In this step, we are going to migrate a database from another Legacy SQL Server having the version 2016, LEGACYSQL2016, which has already been assessed and found fully compatible for potential migration to Azure SQL Managed Instance.

So our objective is now to migrate the database TEAMXX\_GlobalDataDB from LEGACYSQL2016 to SQL MI by using [Azure SQL Managed Instance Feature Link](https://learn.microsoft.com/en-us/azure/azure-sql/managed-instance/managed-instance-link-feature-overview?view=azuresql)

|  |  |  |
| --- | --- | --- |
| **Narrative** | **Screenshot** | **Notes** |
| Open SQL Management Studio and connect to the SQL Server 2016 Source using these details:  **Server:**  **LEGACYSQL2016**  **SQL Authentication**  **Username:**  **Demouser**  **Password:**  **Demo@pass1234567**    Open the ‘Databases’ folder and verify the database to migrate.  **TEAMXX\_GlobalDataDB**  XX is your team number |  |  |
| **Select** the database **TEAMXX\_GlobalDataDB** to replicate  **Right click** **-> Azure SQL Managed Instance Link** -> **Replicate Database** |  | [Replicate a database](https://learn.microsoft.com/en-us/azure/azure-sql/managed-instance/managed-instance-link-use-ssms-to-replicate-database?view=azuresql#replicate-a-database) |
| Click **Next**  This will create a new Managed Instance Link |  |  |
| **Click Next** on the  Requirements check step |  | Prerequisites for source instance:  [Prepare your environment for a link - Azure SQL Managed Instance](https://learn.microsoft.com/en-us/azure/azure-sql/managed-instance/managed-instance-link-preparation?view=azuresql) |
| **Select** your database **TEAMXX\_GlobalDataDB**  Then, **Next** |  |  |
| **Sign In** with you azure user  sqlhackuser**XX**@M365x59576877.onmicrosoft.com |  |  |
| In the Resource Group **SQLHACK-SHARED**, **choose** the SQL MI,  And Log-in to selected Managed Instance with the following credentials:  Demouser  Demo@pass1234567  Then click **Next** |  |  |
| Make sure that the Options for the Distributed Availability Group are set as in the screenshot  And click **Next** |  |  |
| Click **Finish** |  |  |
| All necessary steps will be executed automatically and the Azure SQL MI link will start to replicate data from on-Premise SQL Server 2016 to Azure SQL Managed Instance. |  |  |
| Confirm that your database **TEAMXX\_GlobalDataDB** has been replicated to the SQL Managed Instance:  On your Win10 VM open SQL Management Studio and connect to the target Azure SQL Database Managed Instance using these details:  **Server:**  *(Grabbed at Step 1)*  sqlhackmi-xxxxx.xxxxxxx.database.windows.net  **SQL Authentication**  **Username:**  **Demouser**  **Password:**  **Demo@pass1234567**    Open the ‘Databases’ folder and verify **TEAMXX\_GlobalDataDB** has been replicated and is online. |  |  |

**Failover to Azure SQL Managed Instance**

Execute a failover to Azure SQL Managed Instance using the steps at <https://docs.microsoft.com/en-us/azure/azure-sql/managed-instance/managed-instance-link-use-ssms-to-failover-database?view=azuresql> and remove the configuration AG and DAG only, following the steps below

|  |  |  |
| --- | --- | --- |
| **Select** your database **TEAMXX\_GlobalDataDB on LEGACYSQL2016**  **Right click** on it.  Under Azure SQL Managed Instance link select “**Failover Database”** |  |  |

|  |  |  |
| --- | --- | --- |
| Click “**Next**” |  |  |

|  |  |  |
| --- | --- | --- |
| **Select** your database **TEAMXX\_GlobalDataDB and click “Next”** |  |  |

|  |  |  |
| --- | --- | --- |
| Login with your Azure Account |  |  |

|  |  |  |
| --- | --- | --- |
| Select “**Planned manual failover**” and click “**Next**” |  |  |

|  |  |  |
| --- | --- | --- |
| Select “Delete availability group” on SQL Server  Click “Next” |  | Clean Up is optional.  If you want to keep your databases on your original on-prem SQL Server (LEGACYSQL2016) after failover you can leave the options unchecked. |

|  |  |  |
| --- | --- | --- |
| Click “**Finish**” |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |

Your database has now been migrated to Azure SQL Managed Instance using Managed Instance Link !