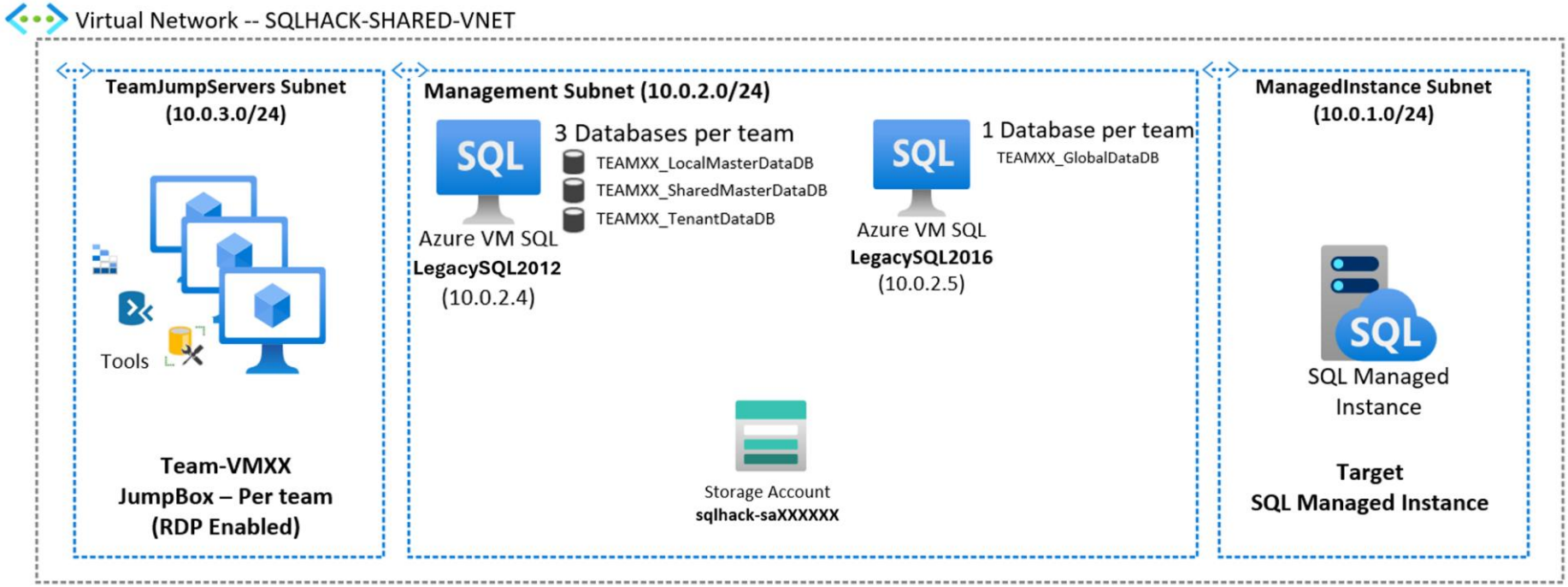
Database Migration Lab Step-by-step (Using Azure Data Studio)

V3.0

Contents

Microsoft

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

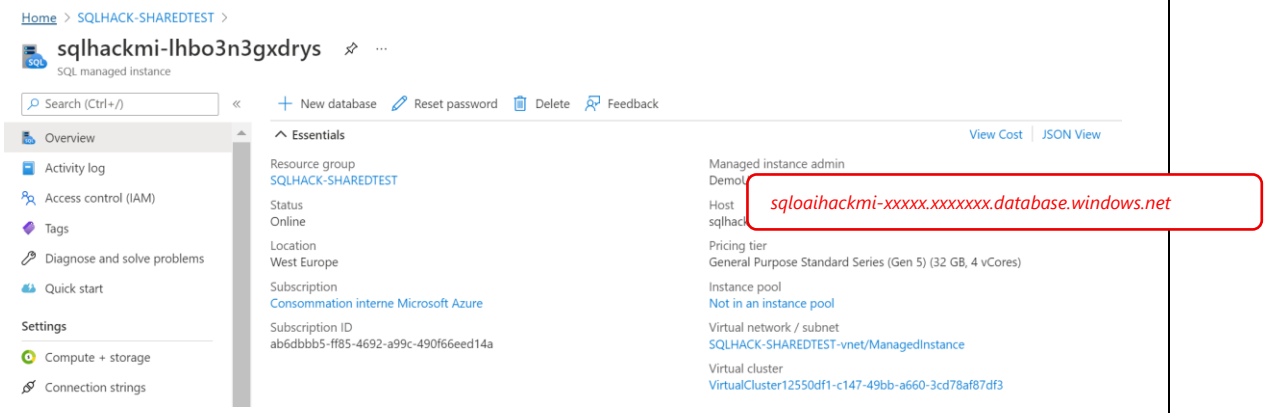
Generic Migration Content

Narrative	Notes
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 DMA & download link: https://docs.microsoft.com/en-us/sql/dma/dma-overview?view=sql-server-ver15 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 Microsoft Migration Portal: https://datamigration.microsoft.com/ 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

Info: All lab documentations can be found here: aka.ms/openaidbhack

1. Get the SQL Managed Instance FQDN

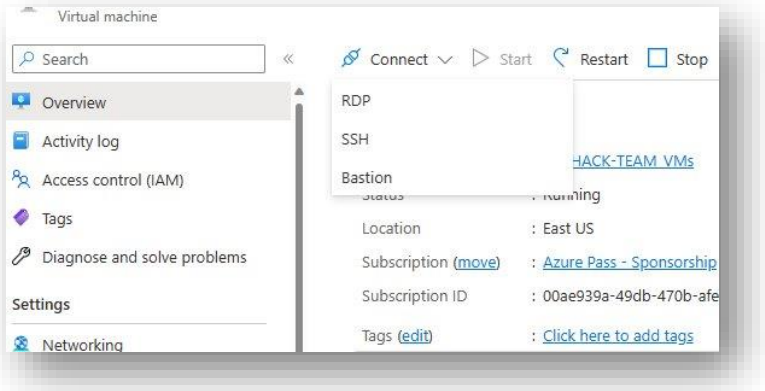
In this section we'll connect to the Azure Portal and retrieve SQL MI information: FQDN, …

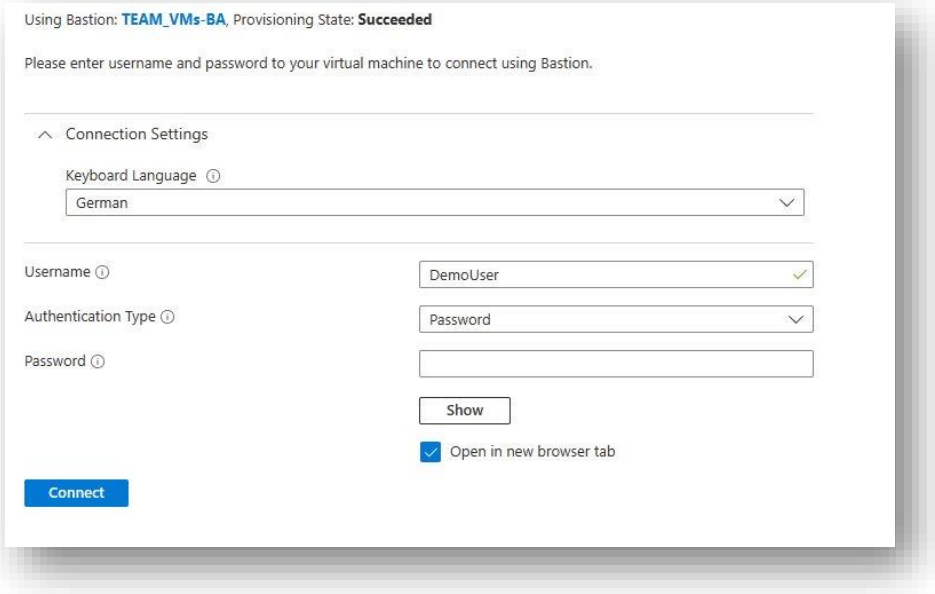
Narrative	Screenshot	Notes
On your Win10 VM open Edge browser and got to the Azure Portal: **HTTPS://portal.azure.com** Log in with your Username and Password: *(see your Teams Group number)* User: **sqlhackuserXX@M365x90478194.onmicrosoft.com** Password: **Ask instructors** In the Azure portal, open the "**SQLOAIHACK-SHARED" Resource Group** and locate the **SQL managed instance** and open it. Note (to a notepad) the **Host Name (FQDN)** in the format: sqloaihackmi-xxxxx.xxxxxxx.database.windows.net Host should be:		

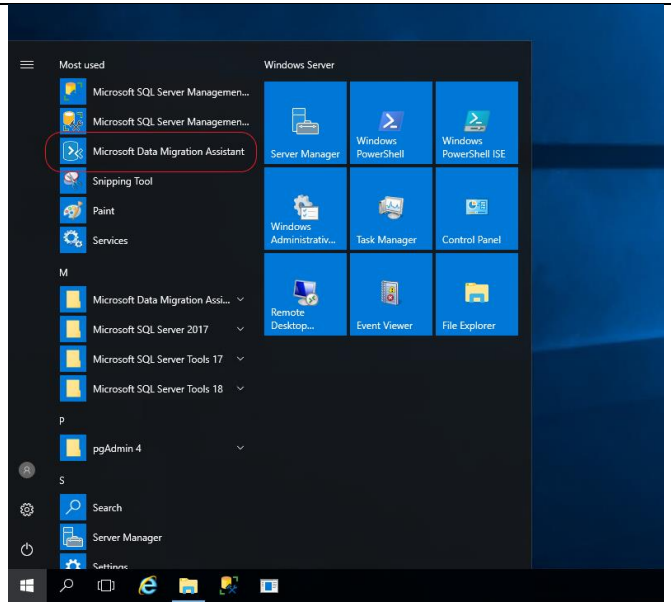
sqlhackmi-k45dnenbp275u.b8df49bc9122.database.windows.net All other **details from the "DB Migration Lab and Parameters.pdf"**		

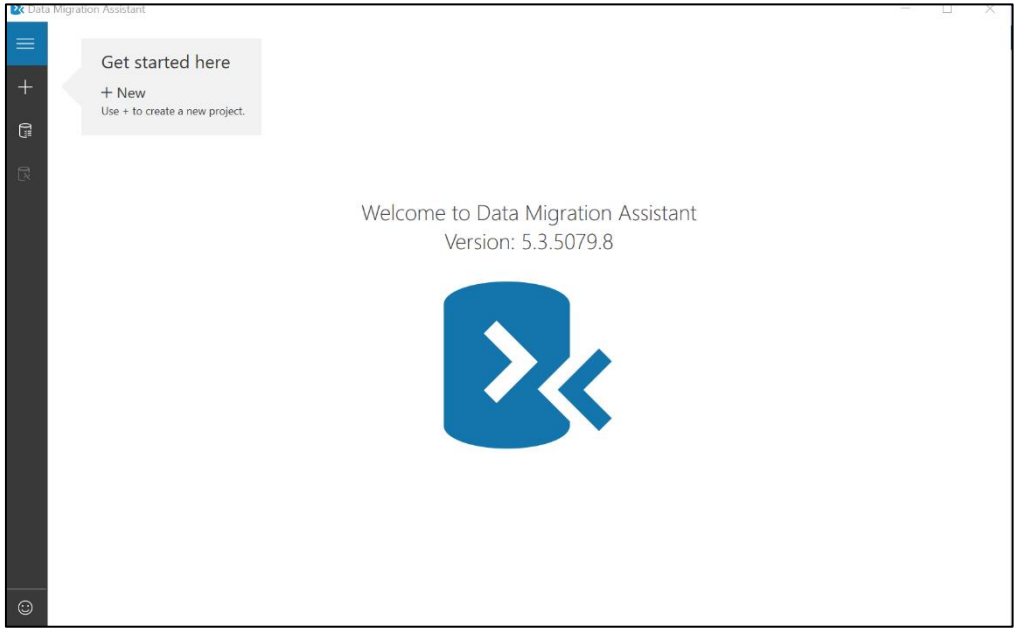
2. Assess the application databases for Azure SQL Database suitability using the Database Migration Assistant (DMA)

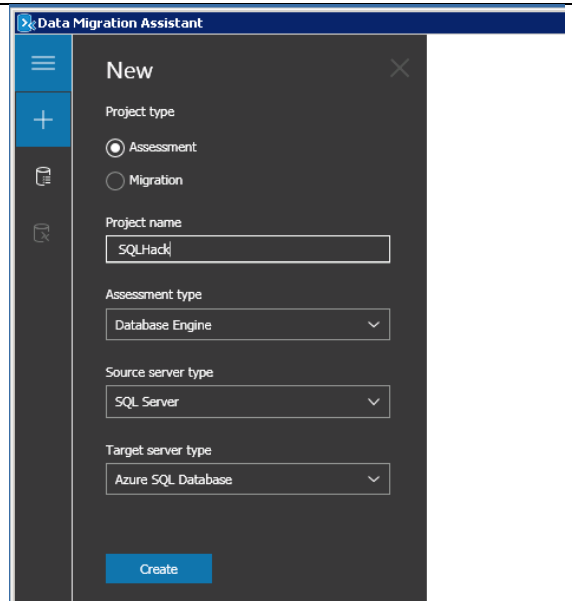
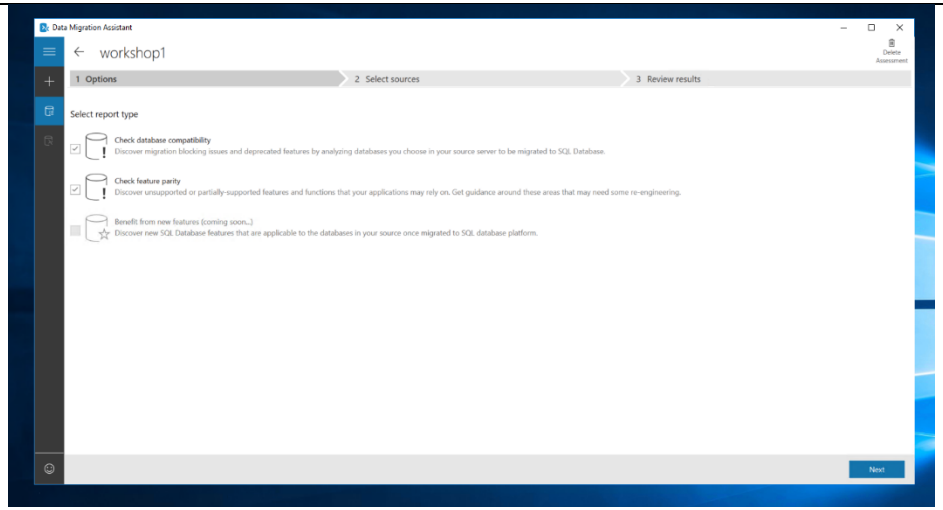
In this section we will use the Data Migration Assistant (DMA) to assess the applications database for suitability for migration to Azure Cloud.

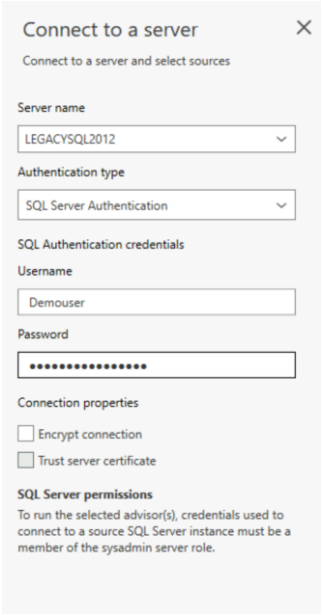
Narrative	Screenshot	Notes
Go to the Azure Portal and navigate to your dedicated Team VM. Click on **Connect > Bastion**. Before connecting, choose **German** as keyboard language. Then use the following credentials (copy to notepad if needed): *Username:* ***DemoUser*** *Password:* ***Demo@pass1234567***		

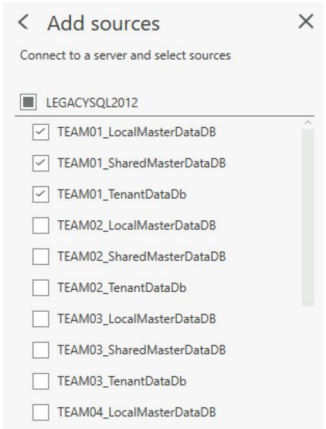
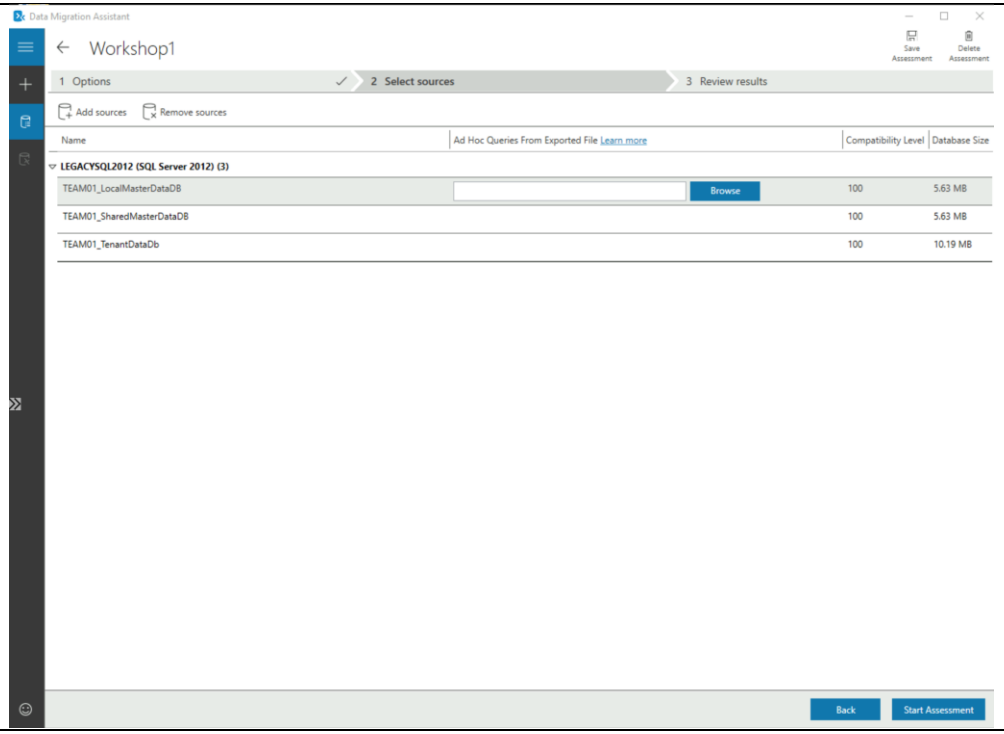

Using Bastion: TEAM_VMs-BA, Provisioning State: Succeeded
Please enter username and password to your virtual machine to connect using Bastion.
Connection Settings
Keyboard Language
German
Username
DemoUser
Authentication Type
Password
Password
Show
Open in new browser tab
Connect

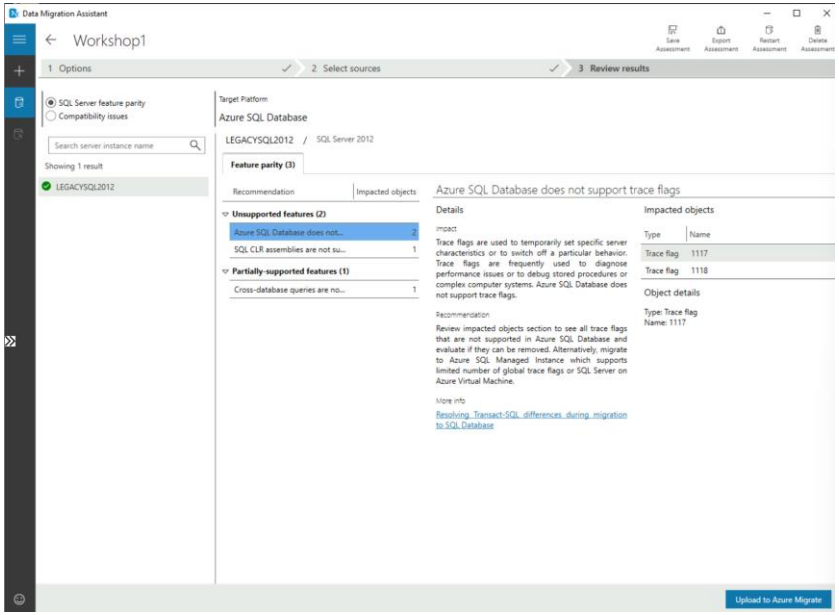
You should have a remote (Bastion) session open to your teams Win10 Management VM. Now run the **Data Migration Assistant** (DMA) from the Start menus or Desktop icon. We need to determine the suitability of the database(s) for migration to Azure. This includes checking for compatibility and feature 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.

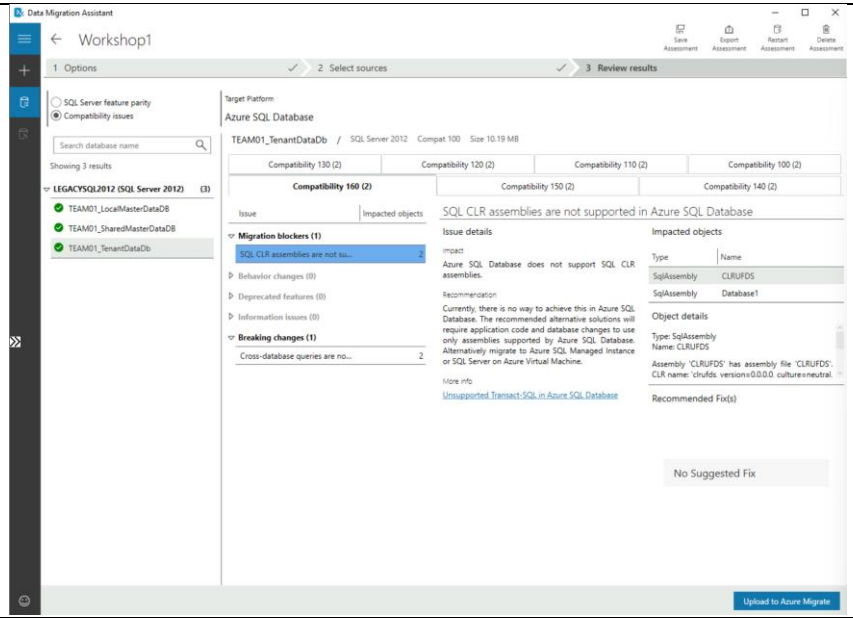
You should see this window open up: Select the **"+"** button to create a **new assessment project**.		

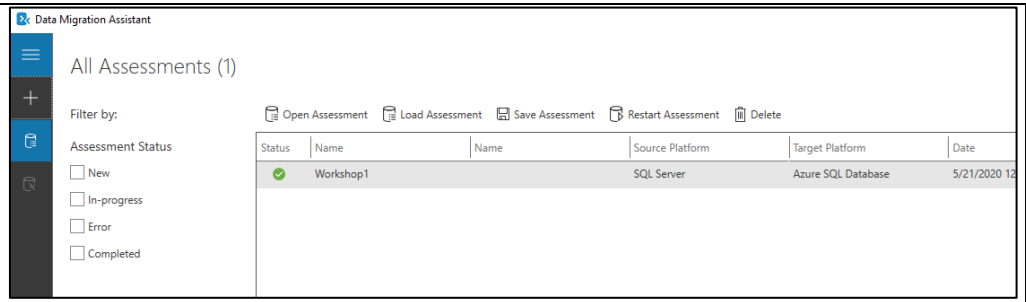
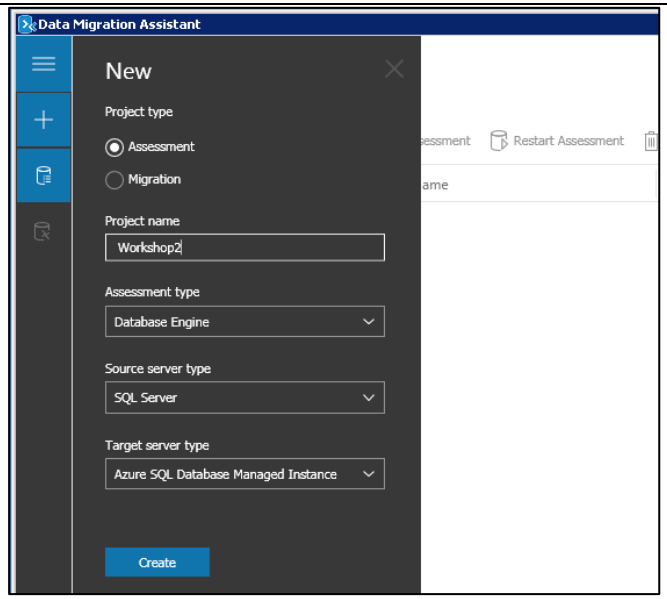
Select/enter the following details in the mask: **Project name:** ***Workshop1*** **Assessment type:** ***Database Engine*** **Source server type:** ***SQL Server*** **Target server type:** ***Azure SQL Database*** Click '**Create'.**		Our first project assessment assumes we will be migrating to Azure 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.

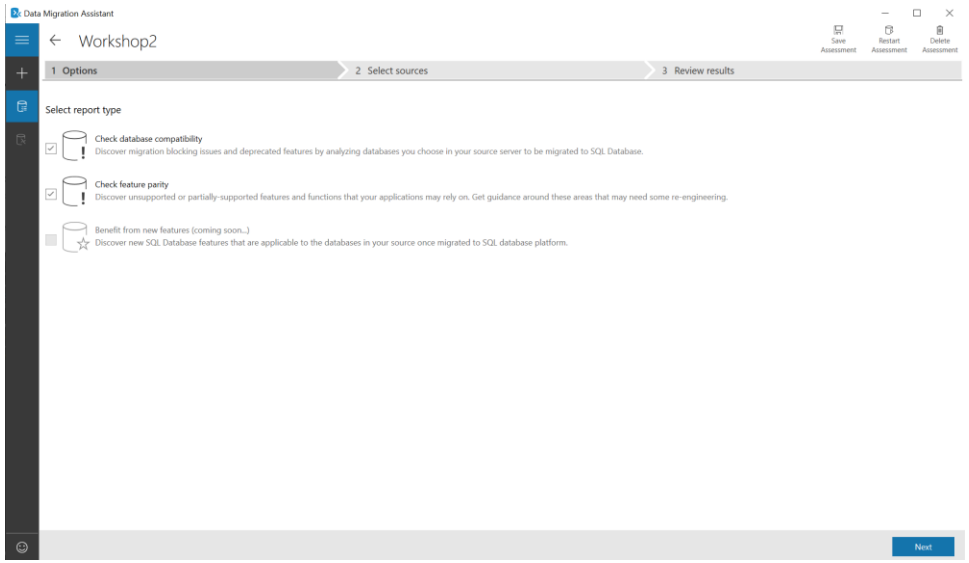
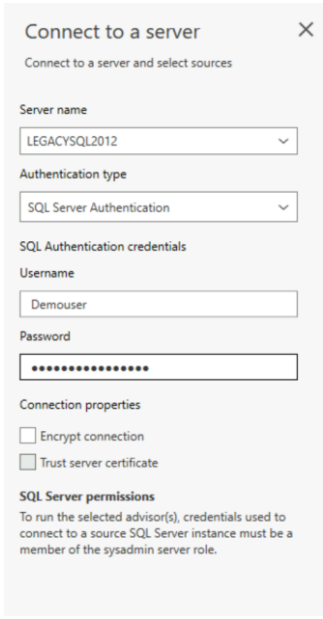
Enter the source/legacy SQL details on the right side: **Server Name:** ***LEGACYSQL2012*** **Authentication Type:** ***SQL Server Authentication*** ***NOTE: Double check that there are no empty spaces in username or password!*** **Username:** ***Demouser*** **Password:** ***Demo@pass1234567*** **Untick "Encrypt connection"** Click '**Connect**' ***If you get an error logging in check 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.

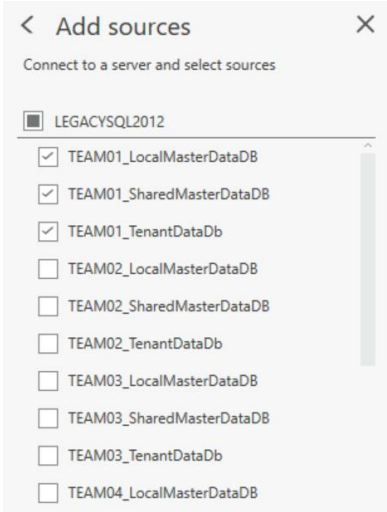
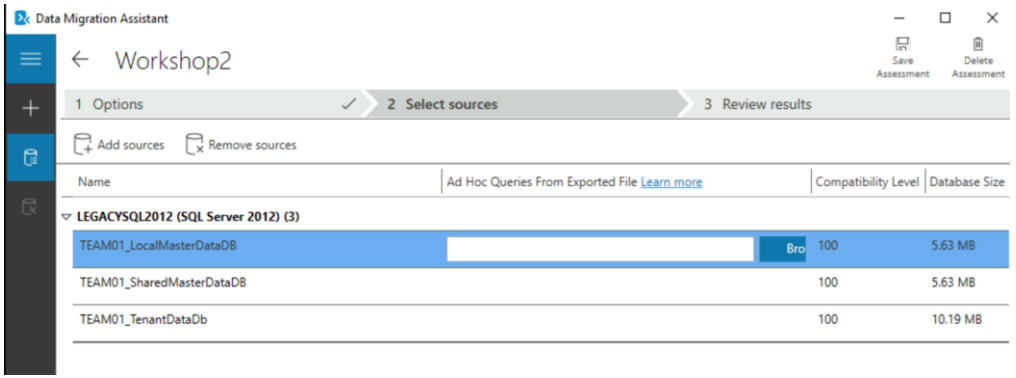
Select **only the 3 databases** used by your 'Online Transaction Monitor' app. These will have a **TEAMxx prefix** where XX should be replaced by **your team number**. ***TEAMxx_LocalMasterDataDb*** ***TEAMxx_SharedMasterDb*** ***TEAMxx_TenantDataDb*** Click '**Add**' to add them to the assessment.		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 3 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

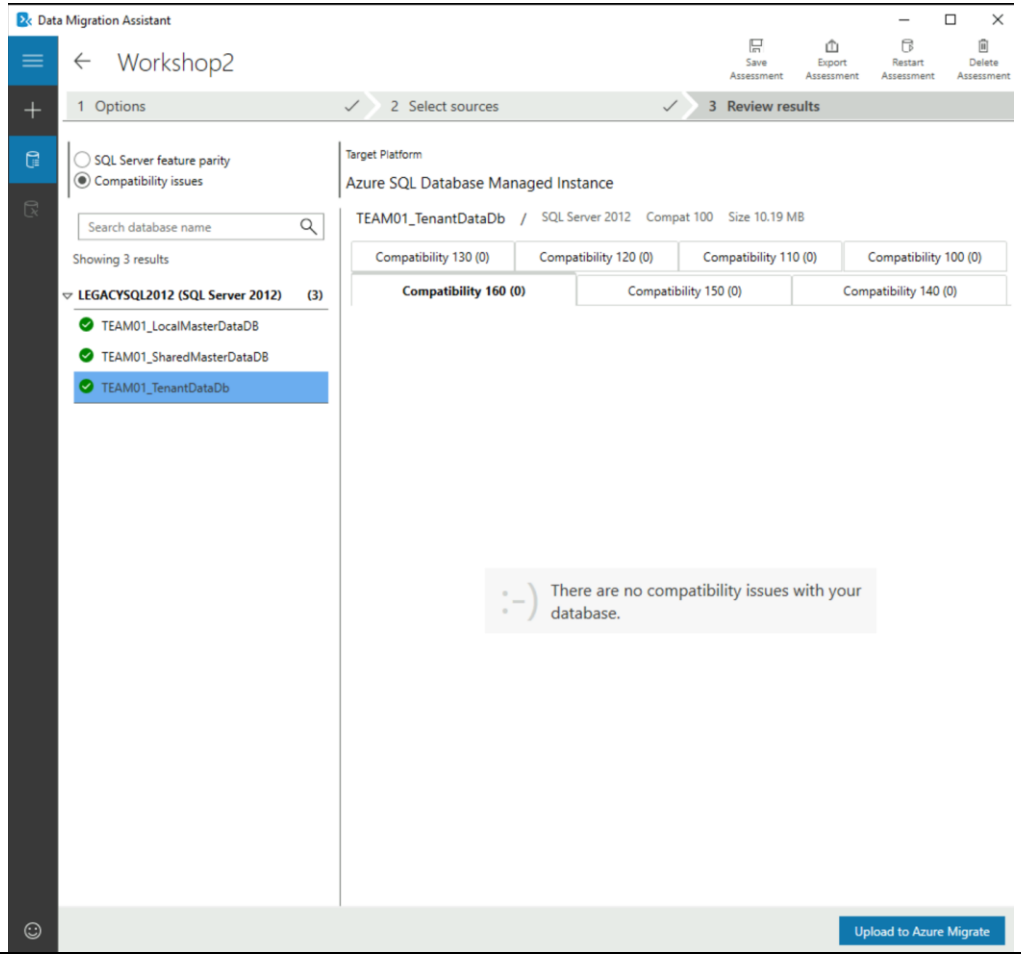
DMA will now show the results of the assessment using 2 separate reports, shown on the left side under “options”: 1. Report: ‘**SQL Server feature parity**’ which is a server level report highlighting any server settings or components (e.g. MSDTC) that the source DBs are using that isn’t supported on 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).

CLR is not supported on Azure SQL DB but is supported by Azure SQL Database Managed Instance (SQLMI).		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 near *Workshop1* Name on the upper left corner to see a list of current DMA projects.		

You should see a view similar to the 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*** ***NOTE: Double check that there are no empty spaces in username or password!*** **Username:** ***Demouser*** **Password:** ***Demo@pass1234567*** **Untick "Encrypt connection"**		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.

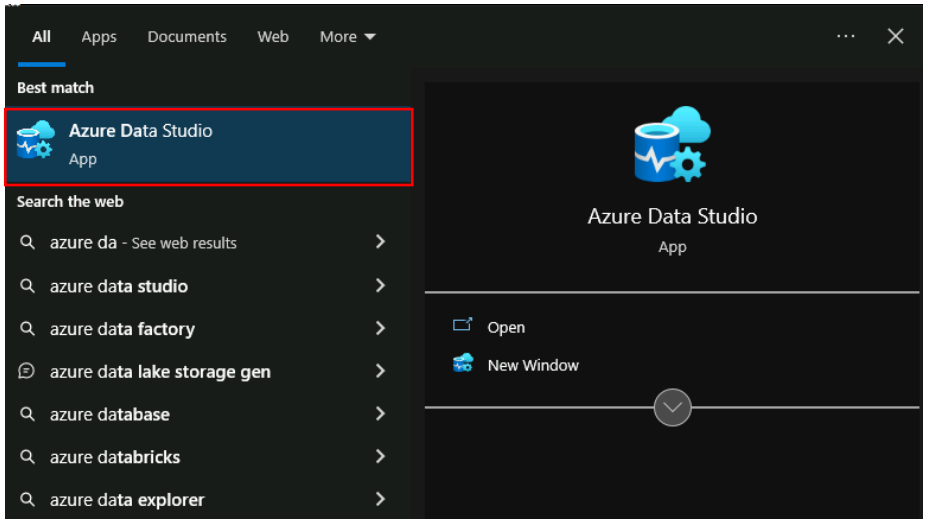
Click '**Connect'.**		
Again, select **only the 3 database** used by your 'Online Transaction Monitor' app. These will have a **TEAMXX prefix** where XX should be **replaced by your team number**. ***TEAMxx_LocalMasterDataDb*** ***TEAMxx_SharedMasterDb*** ***TEAMxx_TenantDataDb*** Click '**Add'** to 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 3 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.

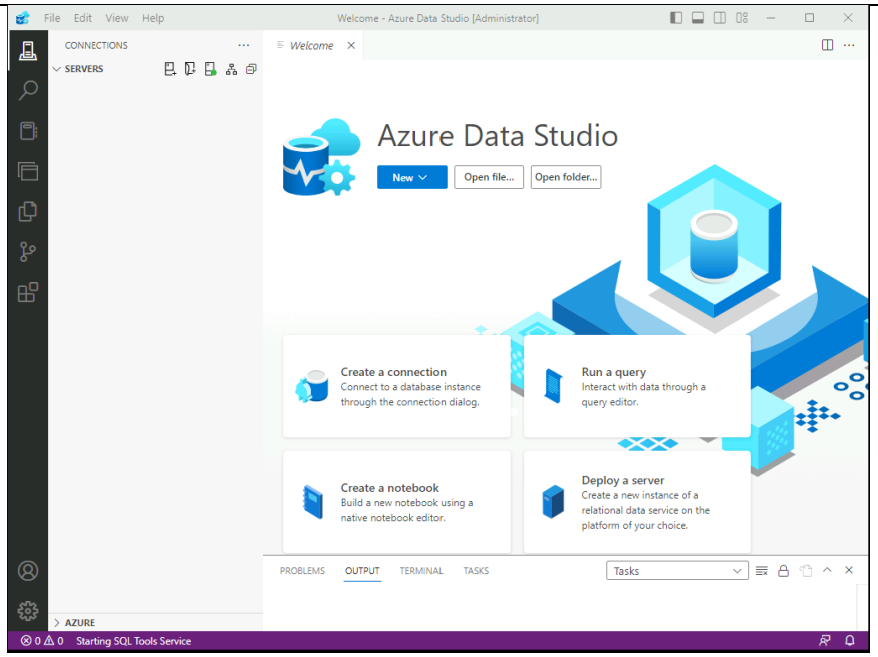
As before, DMA will now show the results from the assessment as the separate 2 reports. Note the '**SQL Server feature parity**' report will either be clean The '**Compatibility Issues**' report should be clear for all 3 databases showing that they can be migrated to Azure SQLDB Managed Instance without 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.

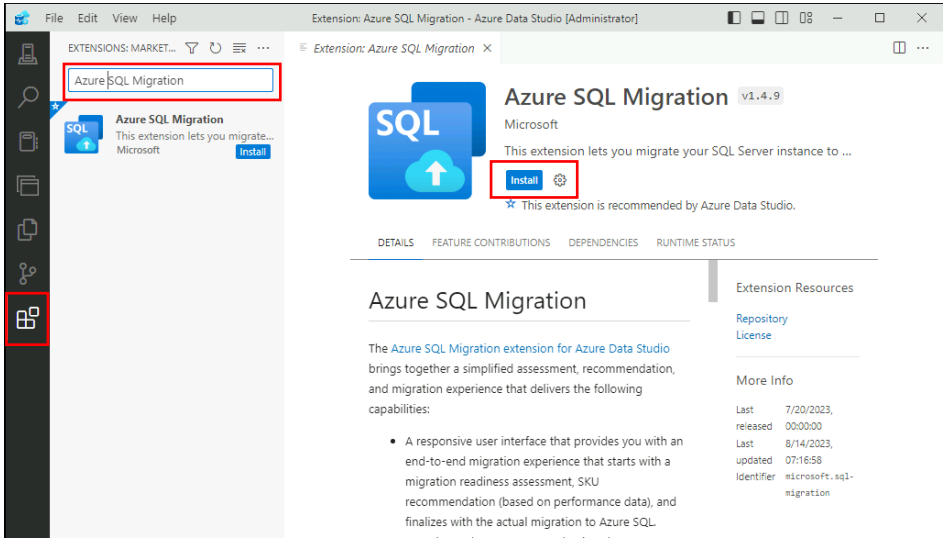
	We are now ready to migrate the application databases to Azure SQL Database Managed Instance	

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
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 an RDP (or Bastion) session open to your teams Win10 Management VM. Now run **Azure Data Studio** (ADS) from the Start menus or Desktop icon.		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. Download and install Azure Data Studio - Azure Data Studio | Microsoft Learn

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: ***sqlhackuserXX@M365x90478194.onmicrosoft.com***

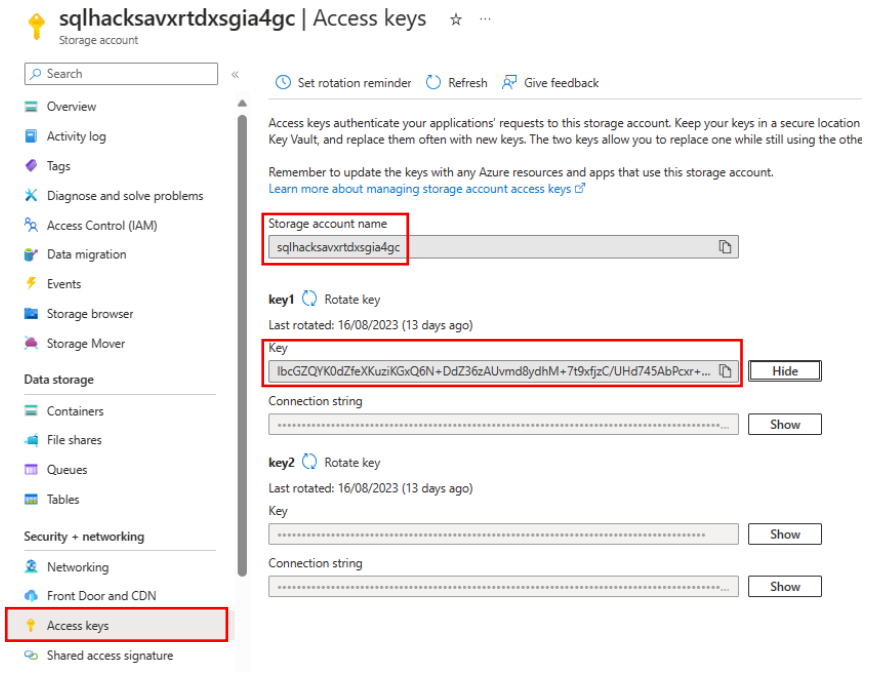
Password: ***Ask instructors***

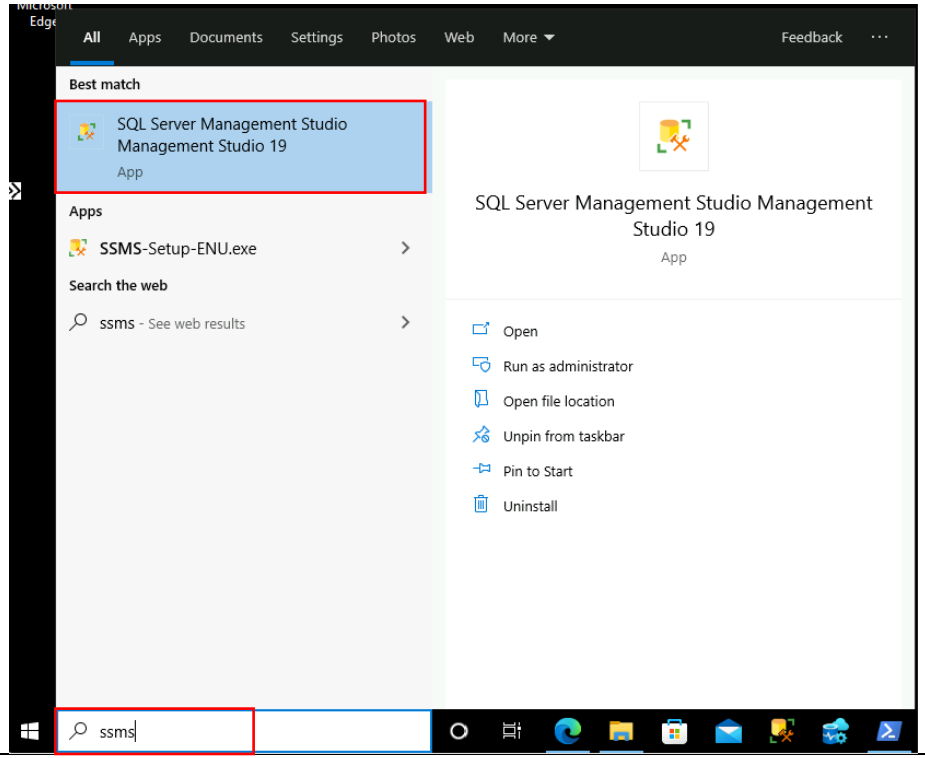
Go to the Resource Group **"SQLOAIHACK-SHARED".**

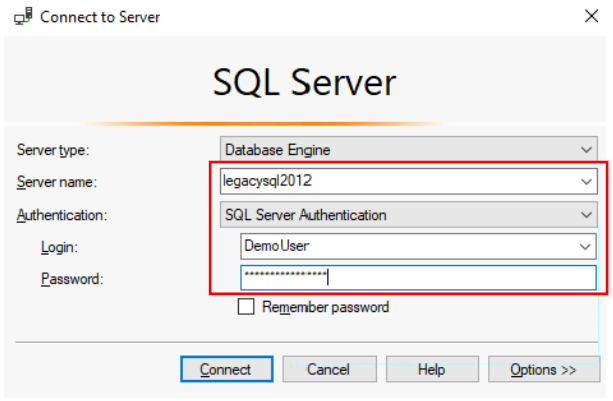
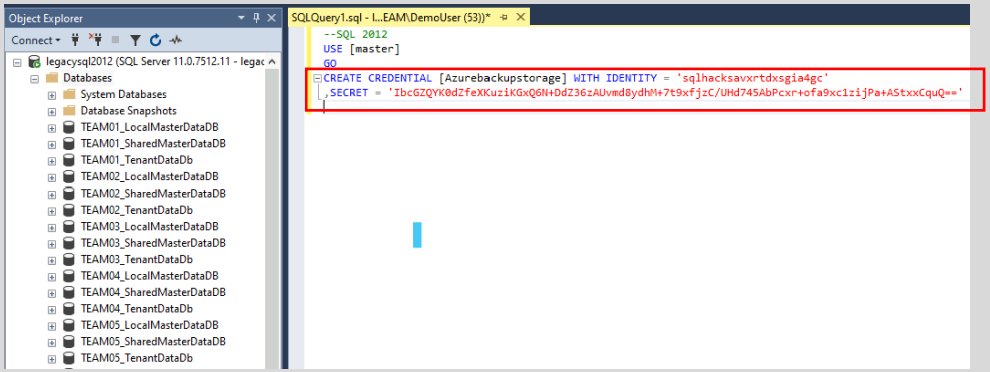
Select the **Azure Storage Account** "sqlhack…". *sqlhacksak45dnenbp275u*

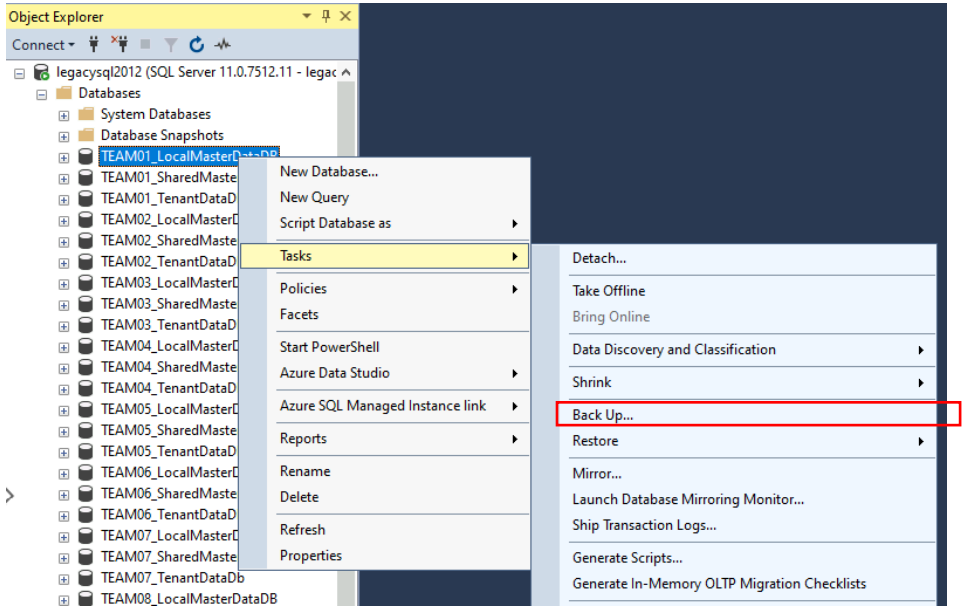
On the left, click: **Access Keys**

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



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.		

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

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>_<databasename>*

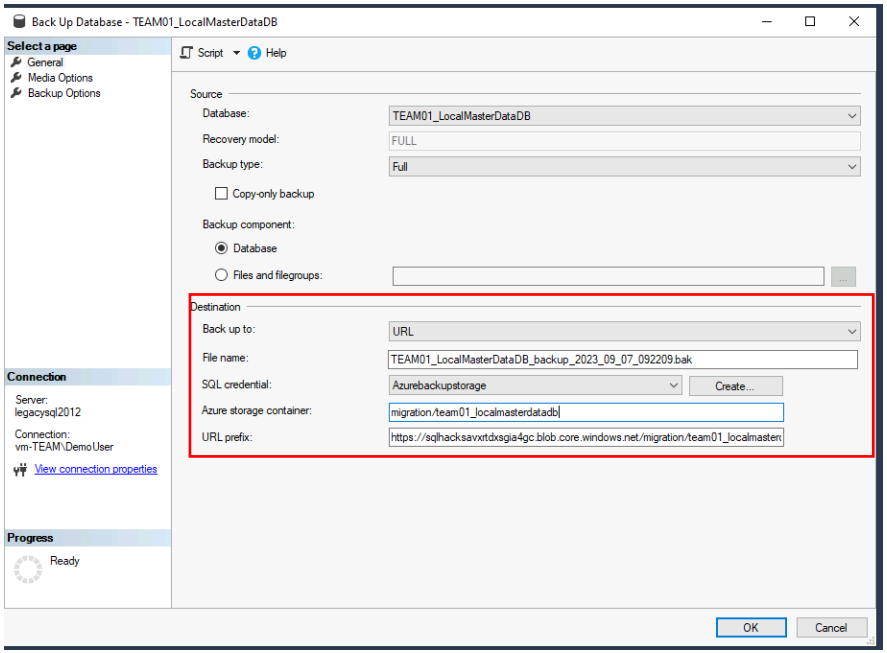
   e.g. for team01 migration/team01_localmasterdatadb

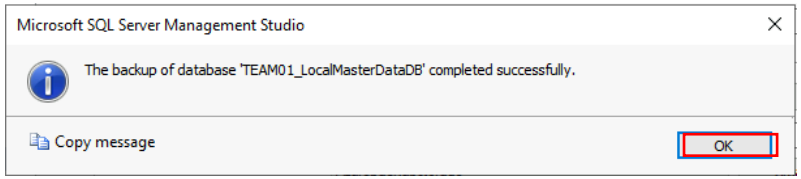
Repeat this process for the remaining 2 databases.

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.

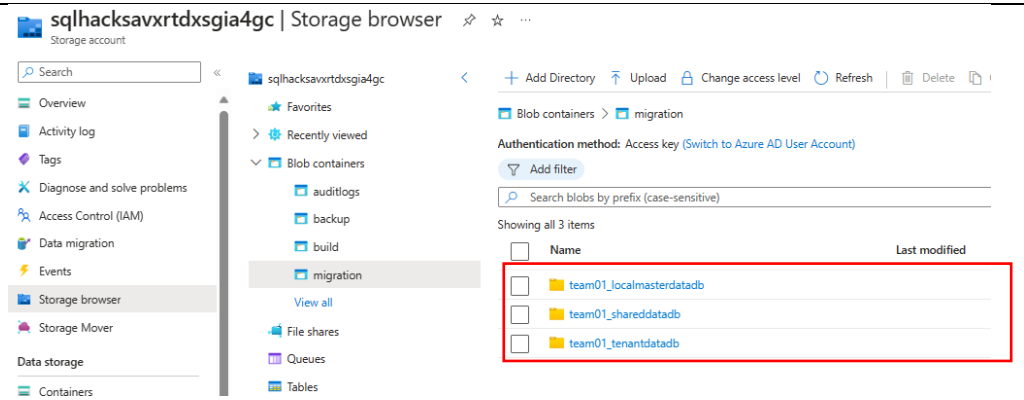
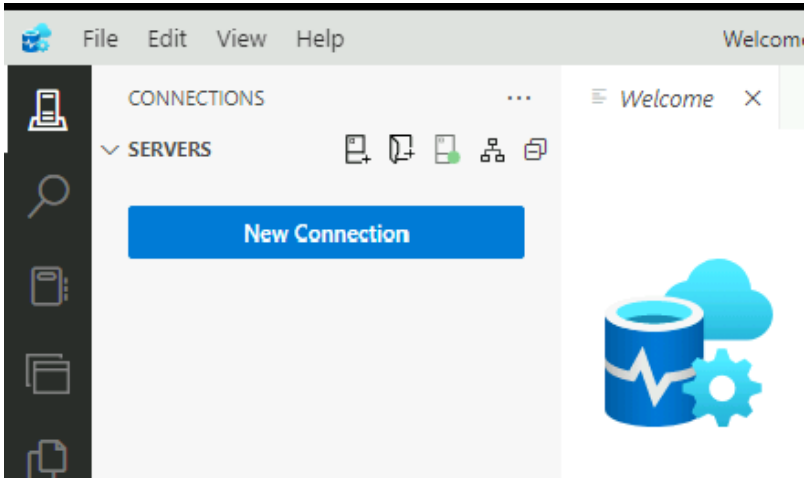


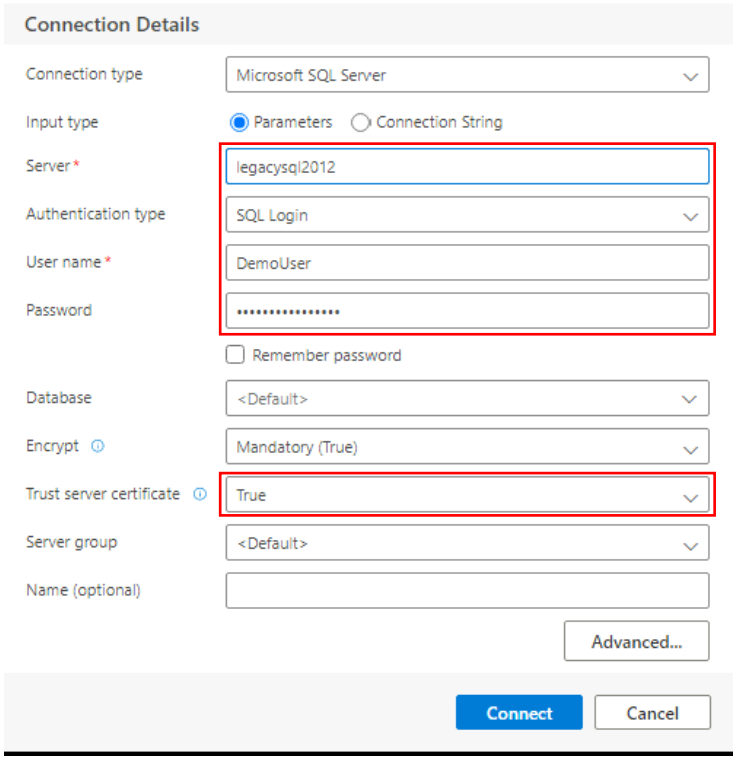


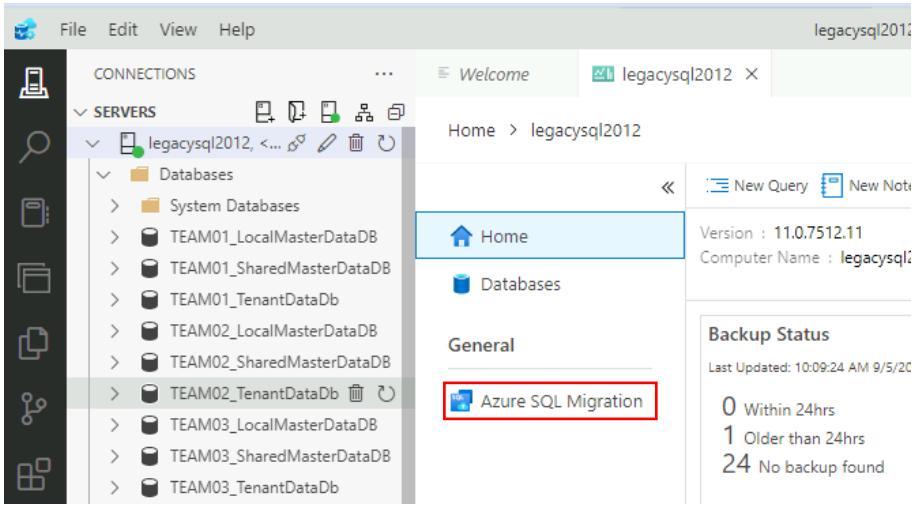
You can also directly use TSQL to BackUp your Databases:

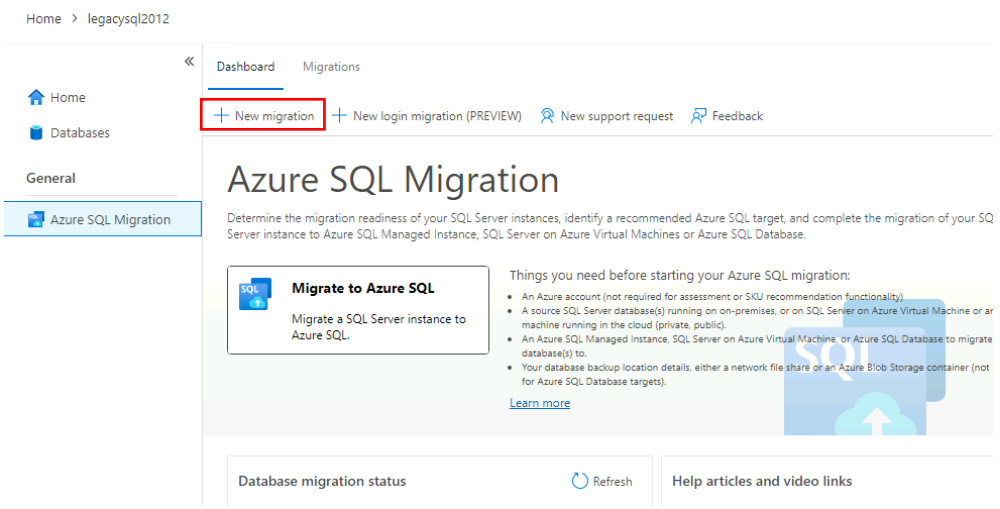
```
BACKUP DATABASE
[TEAM01_LocalMasterDataDB]
TO URL =
N'https://sqlhacksavxrtdxsgia4gc.blob.core.windows.net/migration/team01_localmasterdatadb/TEAM01_LocalMasterDataDB_backup_2023_09_07_092209.bak'
WITH CREDENTIAL = N'Azurebackupstorage',
NOFORMAT, NOINIT, NAME =
N'TEAM01_LocalMasterDataDB-Full Database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

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!***		

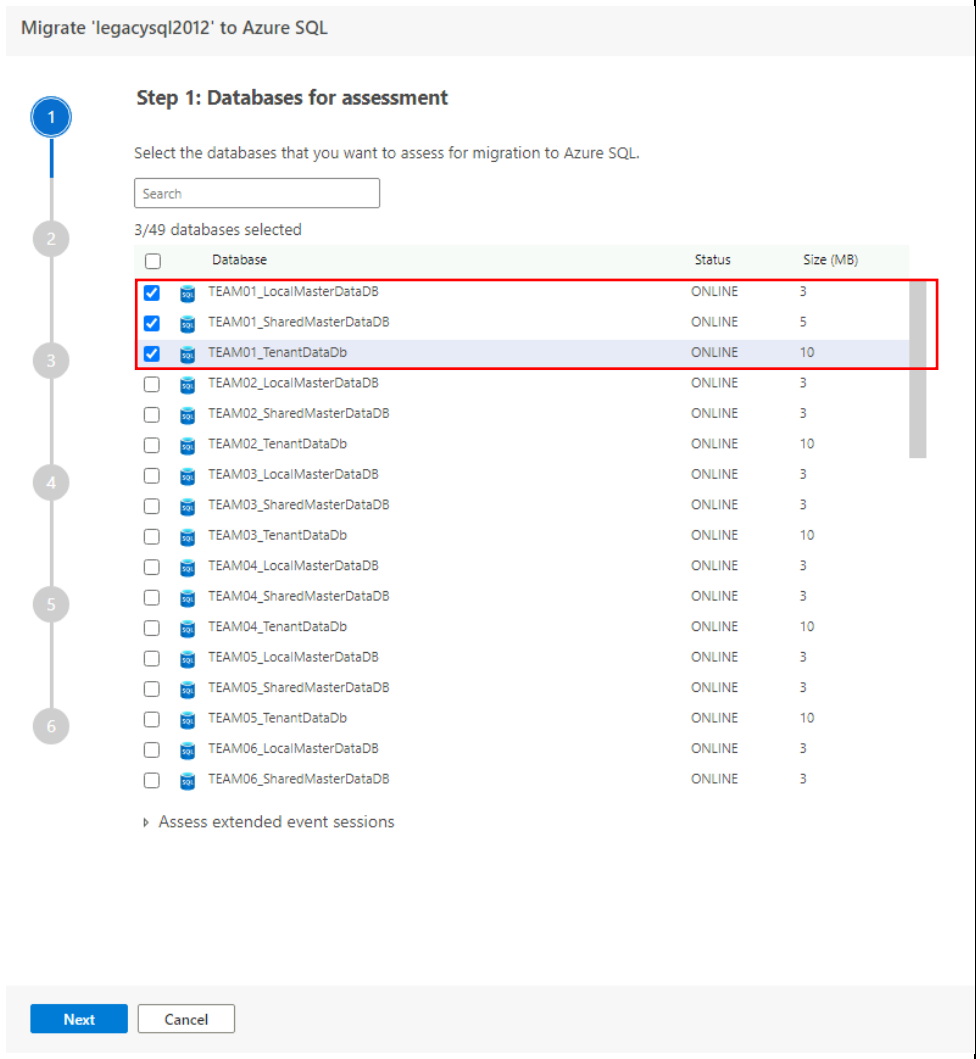
(Optional if you don't see the view yet) Right Click on the SQL Server instance **legacysql2012** on the left-hand side and select "**Manage**". Then click "**Azure SQL Migration**" under General.		

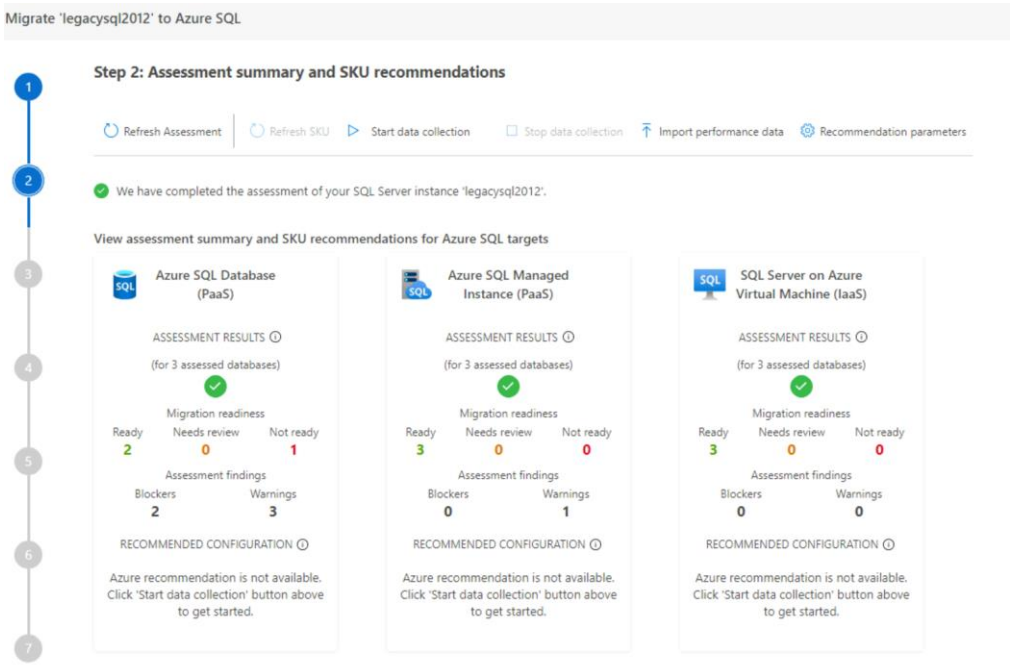
In the tab **Azure SQL migration**, now choose + "**New migration**".		

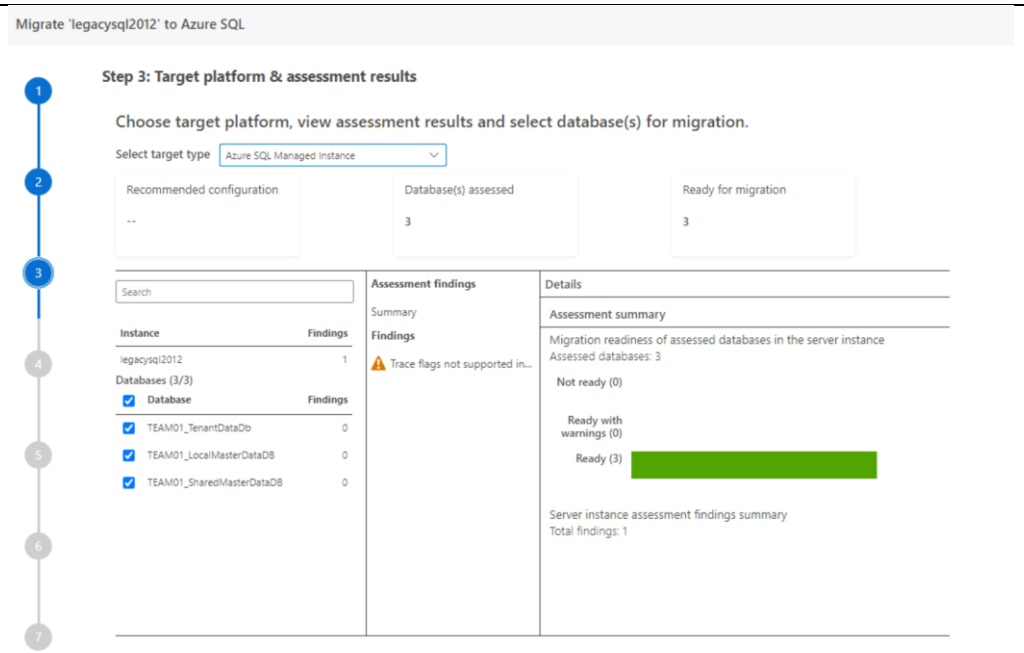
Select **your 3 team databases**:

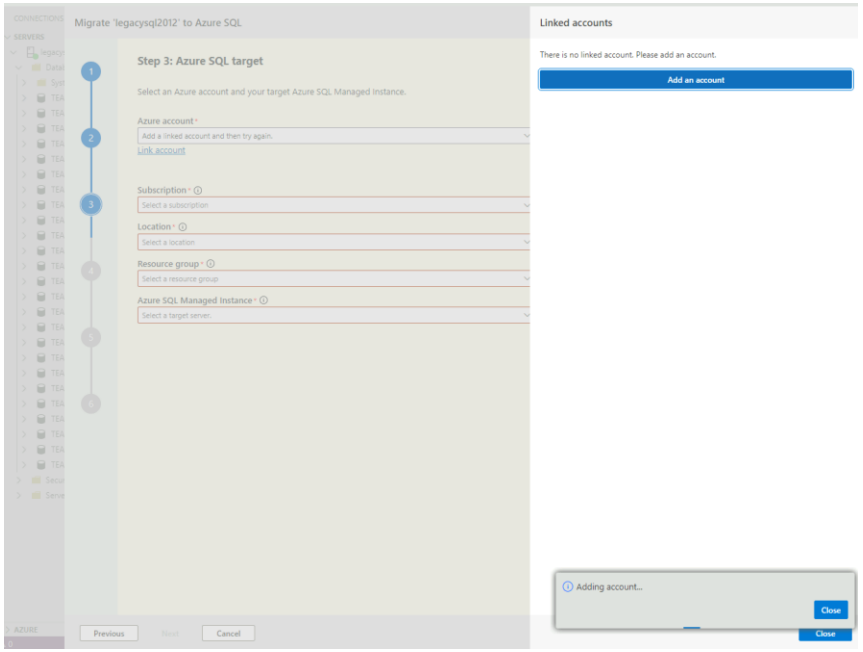
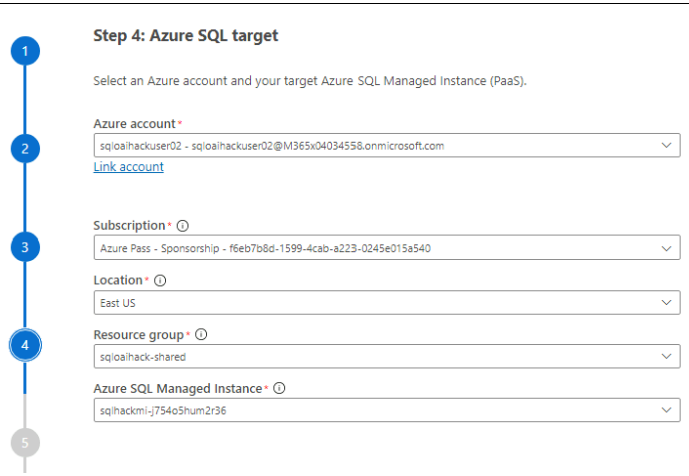
1) TEAMXX_TenantDataDB
2) TEAMXX_LocalMasterDataDB
3) TEAMXX_SharedMasterDataDB

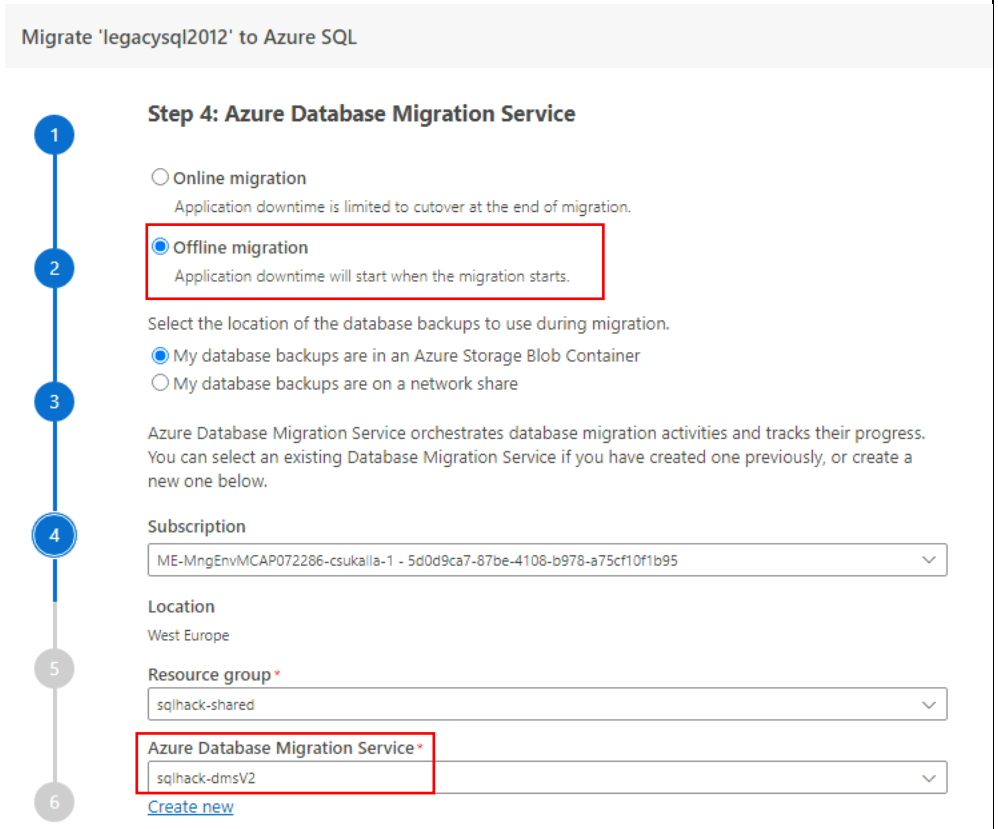
And click **Next**.

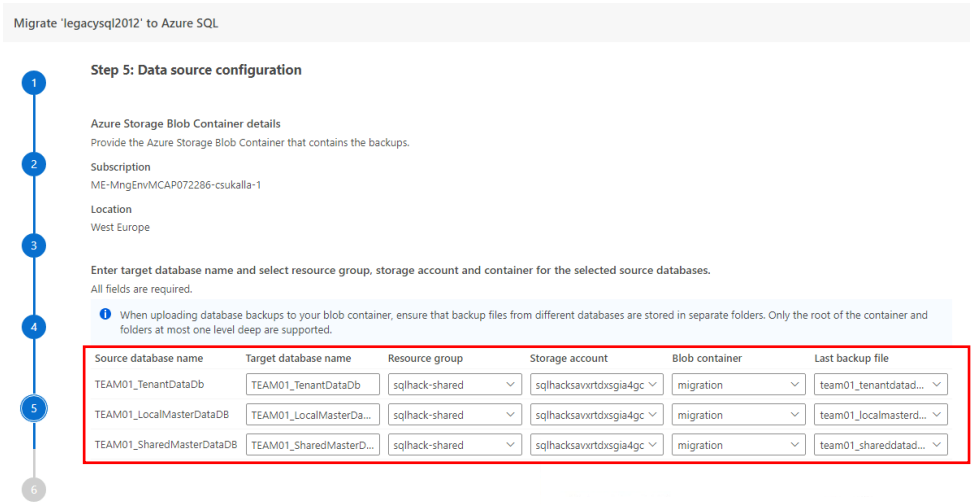


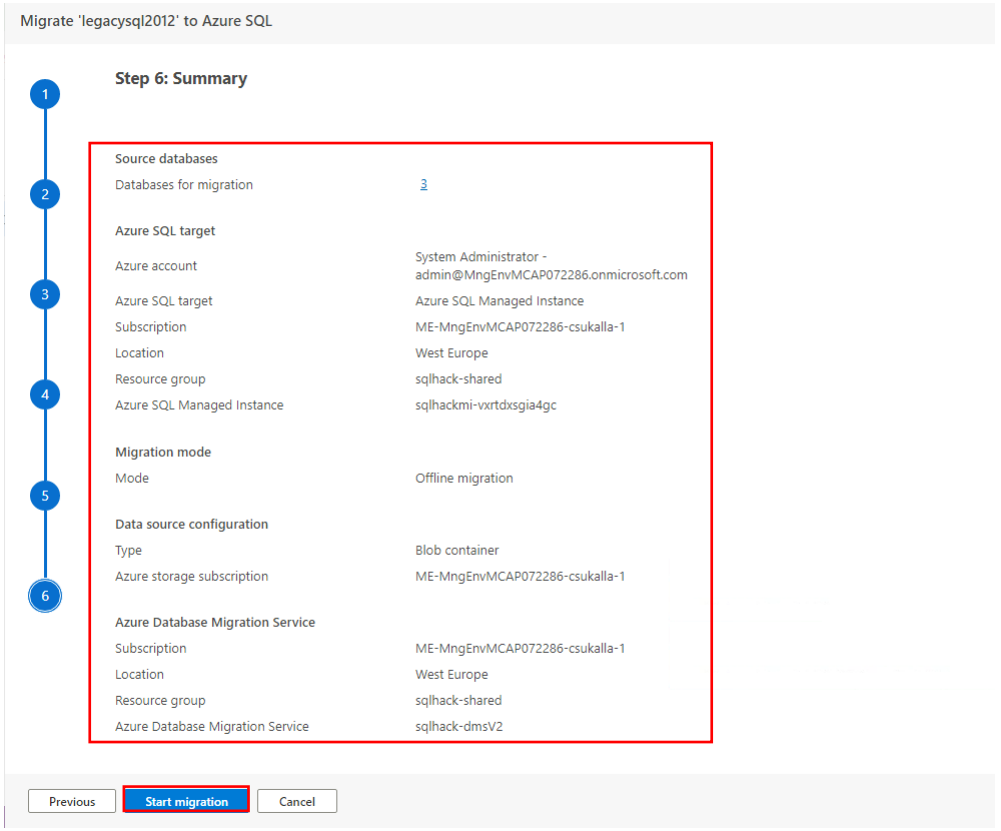
Run assessment and receive recommendations. Click **Next**. Check provided recommendations.		

Select the **target type** as "**Azure SQL Managed Instance**". Click **Next**.		

For this, you need to **add your account** that you use to login to Azure Portal: Similar to… User: ***sqlhackuserXX@ M365x90478194.onmicrosoft.com*** Password: ***Ask instructors***		
Select the **Azure subscription**, the **Location**, **Resource Group** and **Azure SQL MI FDQN Name** which are automatically provided from your Account. And Click **Next**.		

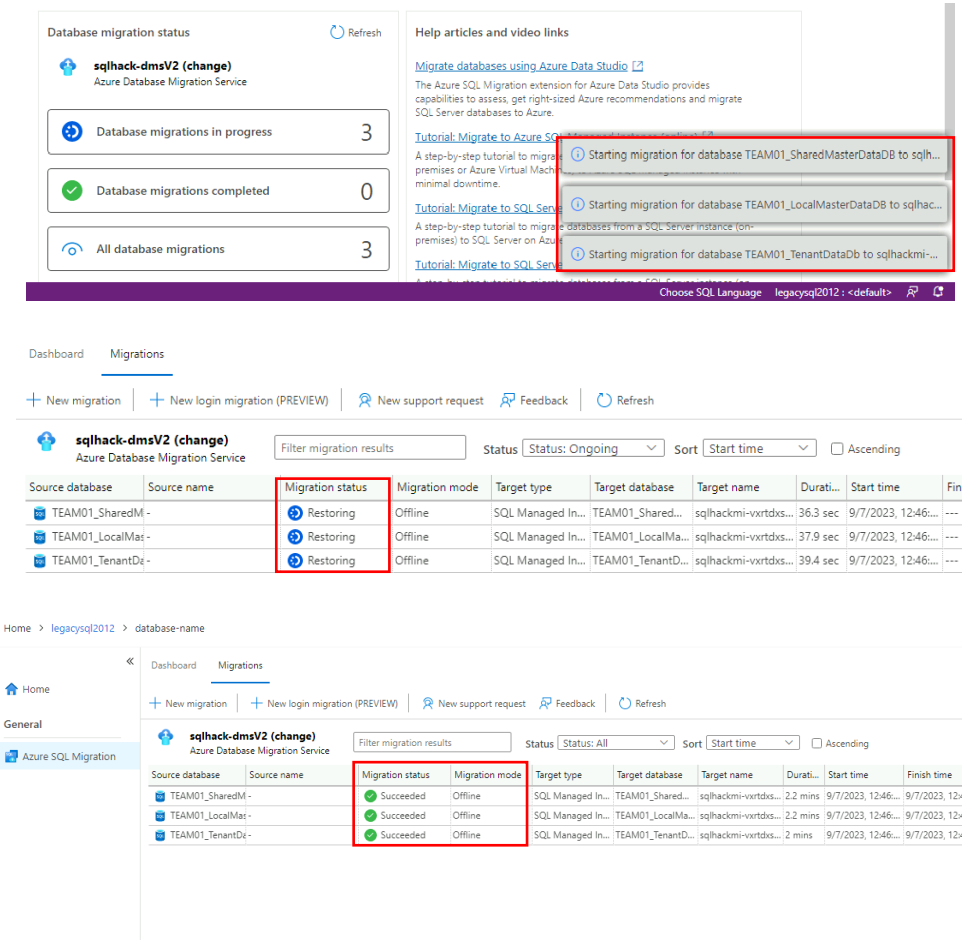
Select “**offline migration**”. Select the existing **Azure Database Migration Service**: ***sqlhack-dmsV2***		You can also do an Online Migration for mission critical workloads using DMS. There are additional steps that you should take for this. Please use the information in the following tutorial for 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.**		

Review summary and **start migration**.		

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.



Microsoft

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

In SSMS connect to the source server: **LEGACYSQL2012** and the target server: **sqloaihackmi-xxxxx.xxxxxxx.database.windows.net** and review the migrated databases. The target server should be: **sqlhackmi-k45dnenbp275u.b8df49bc9122.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!***