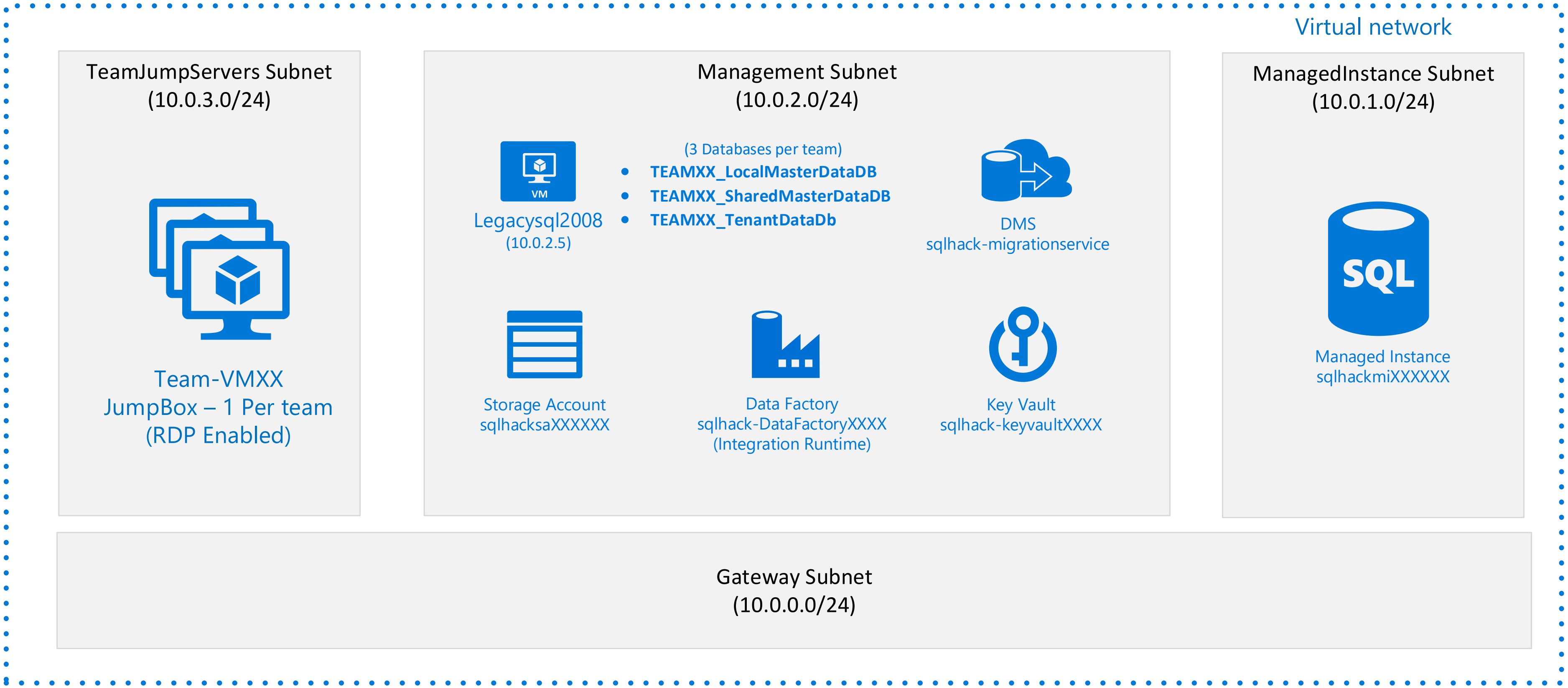
SQL Moderation Hack Database Migration Lab Step-by-step

V2.1

Contents

Migration architecture and Azure components



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

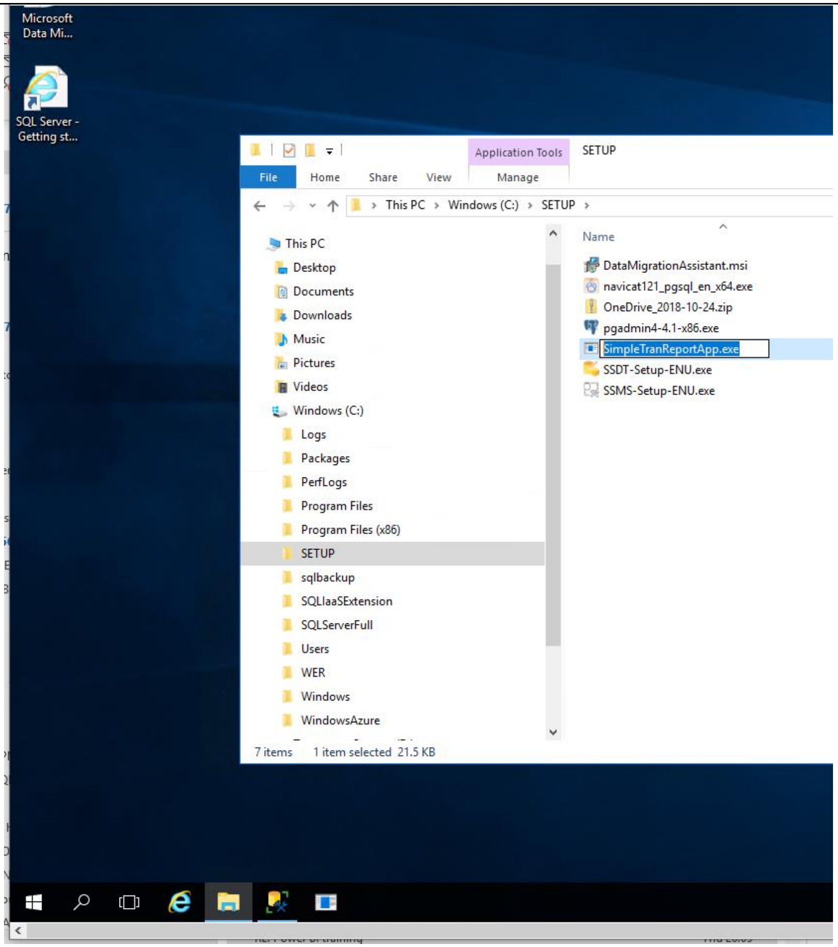
ManagedInstance Subnet
The Azure SQL Managed Instance has been deployed into a dedicated Subnet

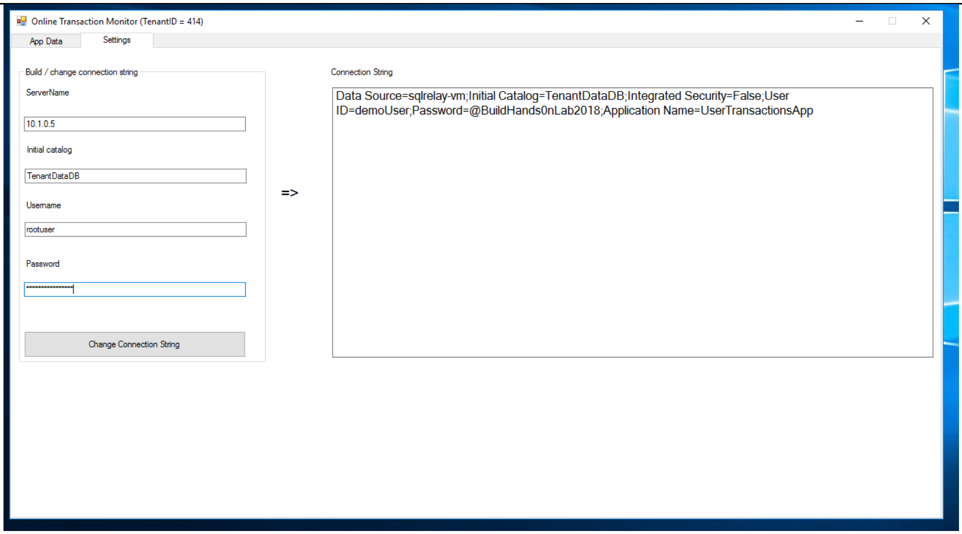
Generic Migration Content

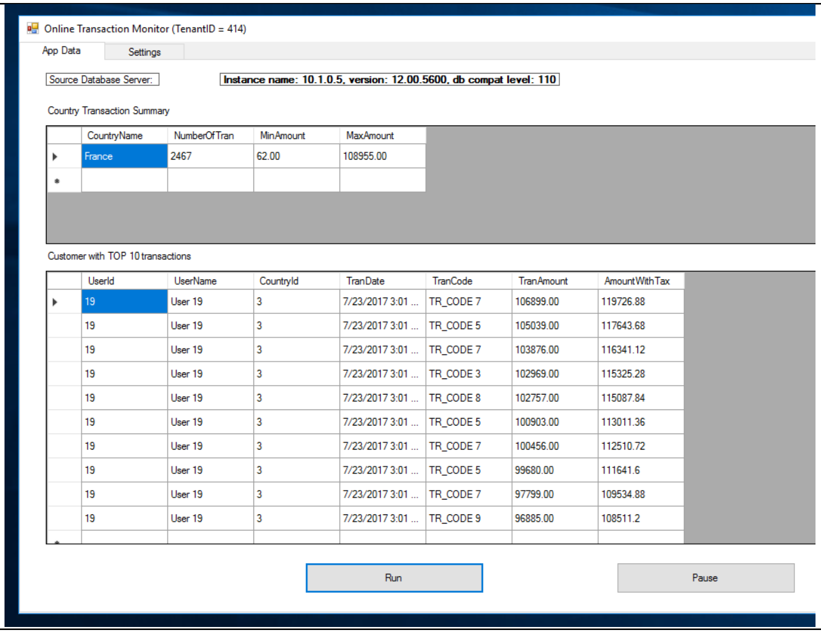
Narrative	Notes
Notes for outside of the workshop: *Familiarise yourself with Microsoft migration tools and the Azure Database Migration Guide*	Azure Database Migration Guide: https://www.microsoft.com/en-us/download/default.aspx DMA & download link: https://docs.microsoft.com/en-us/sql/dma/dma-overview?view=sql-server-2017 Microsoft Migration Portal: https://datamigration.microsoft.com/

1. Start the ‘Online Transaction Monitor’ application

In this section we’ll connect the legacy Online Transaction Monitor application to the legacy SQL2008 databases and see it running.

Narrative	Screenshot	Notes
We will set the sample application running to demonstrate how Azure Database Migration Services can be used to perform a migration of a database. RDP onto the Win10 management VM using the **details from the “DB Migration Lab and Parameters.pdf”** Run the ‘SimpleTranReportApp’ app which an be found in: **C:_SQLHACK_\LABS\01-Data_Migration** Or In Windows Explorer search for ‘**SimpleTranReportApp.exe**’	 Microsoft Data Mi... SQL Server - Getting st... Application Tools SETUP File Home Share View Manage This PC › Windows (C:) › SETUP › This PC Desktop Documents Downloads Music Pictures Videos Windows (C:) Logs Packages PerfLogs Program Files Program Files (x86) SETUP sqlbackup SQLIaaSExtension SQLServerFull Users WER Windows WindowsAzure Name DataMigrationAssistant.msi navicat121_pgsql_en_x64.exe OneDrive_2018-10-24.zip pgadmin4-4.1-x86.exe SimpleTranReportApp.exe SSDT-Setup-ENU.exe SSMS-Setup-ENU.exe 7 items 1 item selected 21.5 KB	In this scenario the legacy app has lost its source code, so only exists as an executable. We are not, however, blocked from migrating to Azure.

Once running, select the '**Settings**' tab and enter the following parameters into the fields identified: **ServerName:** **LEGACYSQL2008** **Initial Catalog:** **TEAMXX_TenantDataDb** **Username:** **TEAMXX** **Password:** **TEAMXX** Click the "**Change Connection String"** button to apply the connection string modifications	 Online Transaction Monitor (TenantID = 414) App Data Settings Build / change connection string ServerName: 10.1.0.5 Initial catalog: TenantDataDB Username: rootuser Password: •••••••••••••••• Change Connection String => Connection String Data Source=sqlrelay-vm;Initial Catalog=TenantDataDB;Integrated Security=False;User ID=demoUser;Password=@BuildHands0nLab2018;Application Name=UserTransactionsApp	Use the parameters from the Appendix in the "Hands-on Lab - Data Migration" document. The connection string will now have been set to connect to the legacy SQL host – **LEGACYSQL2008** with appropriate Team database and login details.

Select '**App Data**' tab and click the "**Run**" button. After a few seconds transaction will start to appear in the application.		The application will generate simulated transactional data. Notice how the 'Source Database Server' information at the top of the app reflects the parameters given in the previous step.

Online Transaction Monitor (TenantID = 414)

Source Database Server: **Instance name: 10.1.0.5, version: 12.00.5600, db compat level: 110**

Country Transaction Summary

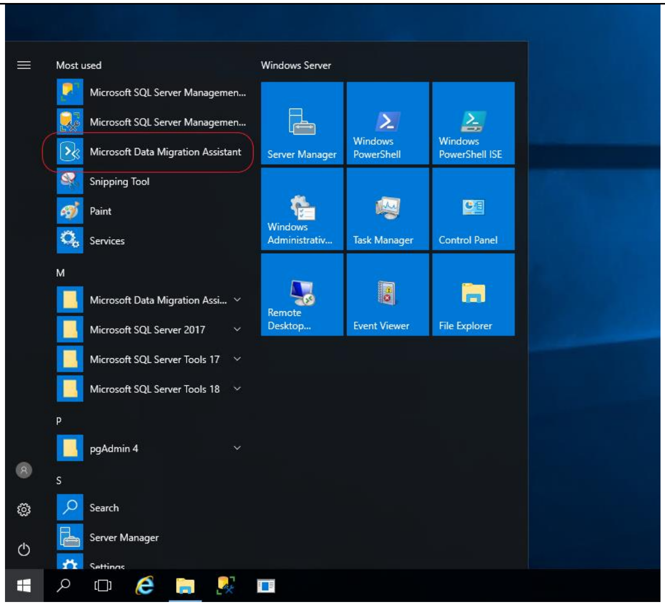
CountryName	NumberOfTran	MinAmount	MaxAmount
France	2467	62.00	108955.00

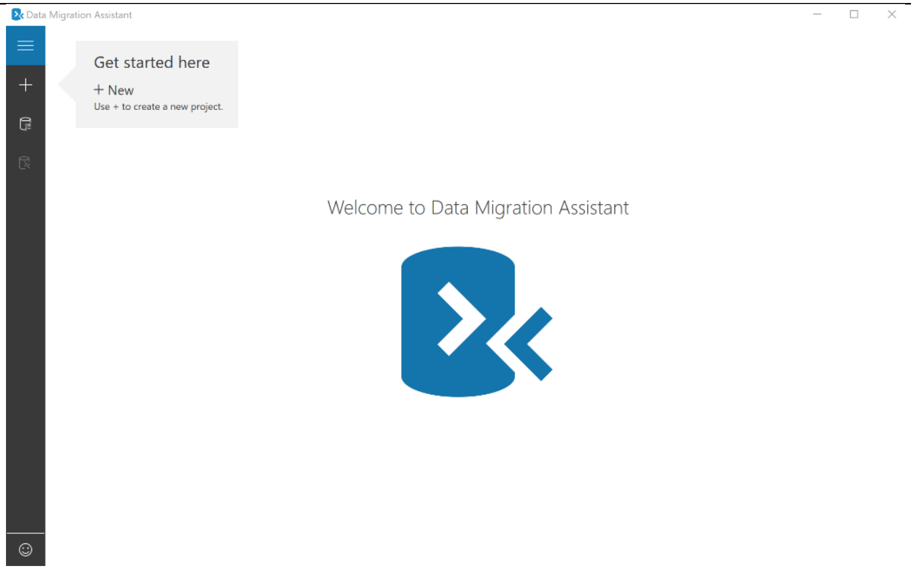
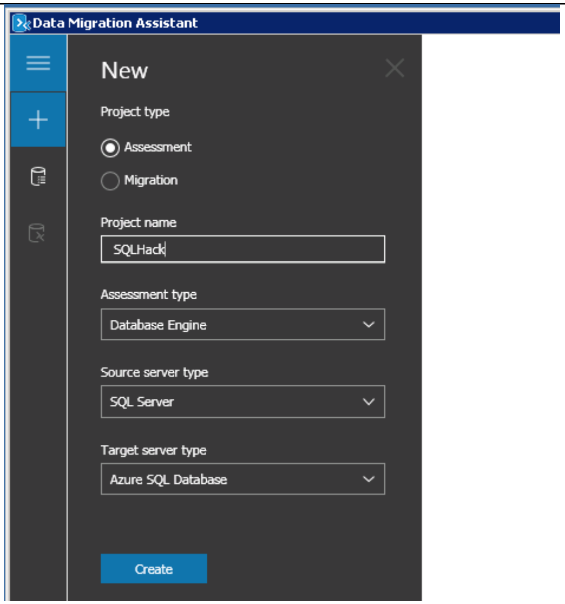
Customer with TOP 10 transactions

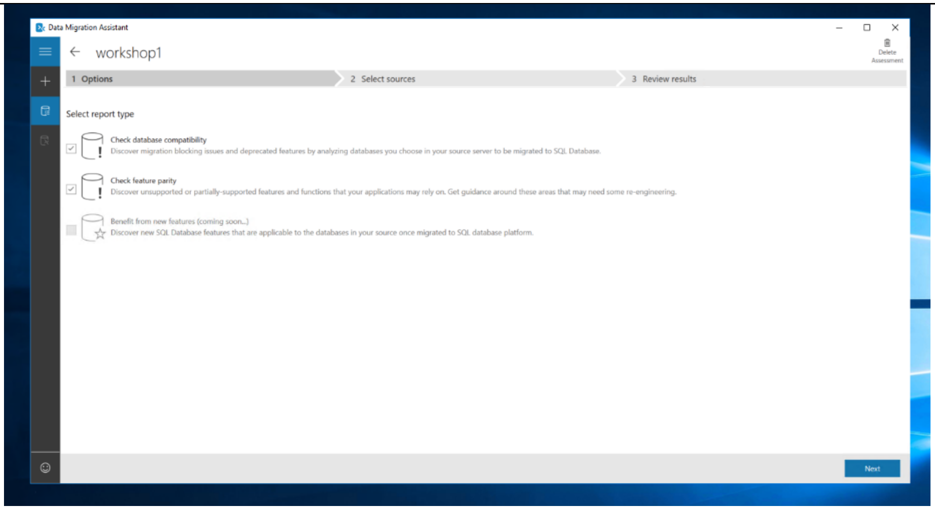
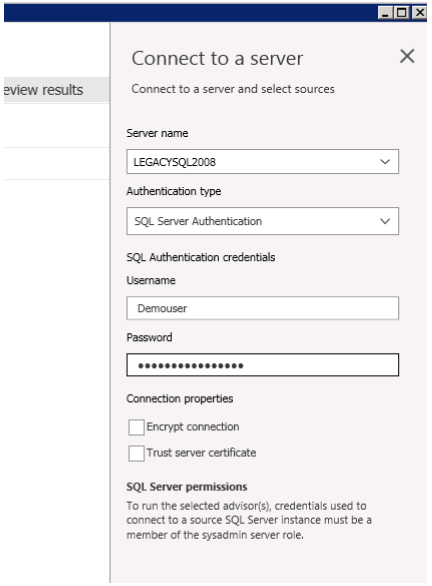
UserId	UserName	CountryId	TranDate	TranCode	TranAmount	AmountWithTax
19	User 19	3	7/23/2017 3:01 ...	TR_CODE 7	106899.00	119726.88
19	User 19	3	7/23/2017 3:01 ...	TR_CODE 5	105039.00	117643.68
19	User 19	3	7/23/2017 3:01 ...	TR_CODE 7	103876.00	116341.12
19	User 19	3	7/23/2017 3:01 ...	TR_CODE 3	102969.00	115325.28
19	User 19	3	7/23/2017 3:01 ...	TR_CODE 8	102757.00	115087.84
19	User 19	3	7/23/2017 3:01 ...	TR_CODE 5	100903.00	113011.36
19	User 19	3	7/23/2017 3:01 ...	TR_CODE 7	100456.00	112510.72
19	User 19	3	7/23/2017 3:01 ...	TR_CODE 5	99680.00	111641.6
19	User 19	3	7/23/2017 3:01 ...	TR_CODE 7	97799.00	109534.88
19	User 19	3	7/23/2017 3:01 ...	TR_CODE 9	96885.00	108511.2

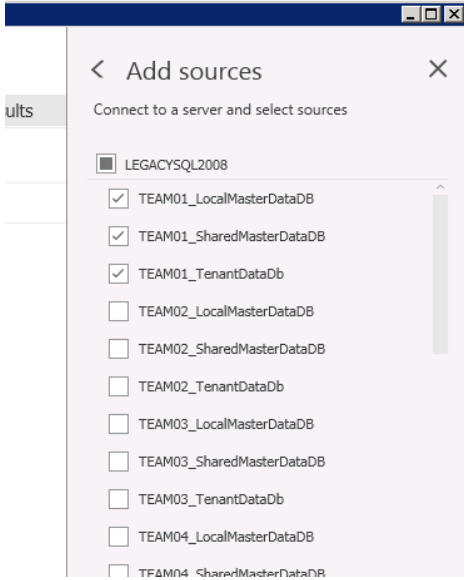
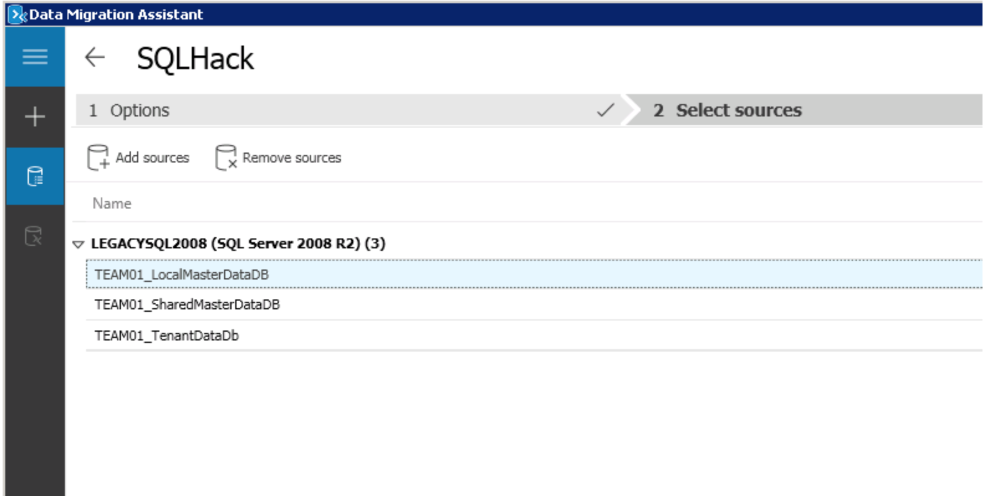
2. Assess the application databases for Azure SQL Database suitability

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 an RDP session open to your teams Win10 Management VM, if so run DMA from the Start menus or Desktop icon.	 Most used Microsoft SQL Server Managemen... Microsoft SQL Server Managemen... Microsoft Data Migration Assistant Snipping Tool Paint Services M Microsoft Data Migration Assi... Microsoft SQL Server 2017 Microsoft SQL Server Tools 17 Microsoft SQL Server Tools 18 P pgAdmin 4 S Search Server Manager Settings Windows Server Server Manager Windows PowerShell Windows PowerShell ISE Windows Administrativ... Task Manager Control Panel Remote Desktop... Event Viewer File Explorer	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	 Data Migration Assistant Get started here + New Use + to create a new project. Welcome to Data Migration Assistant	
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**'	 Data Migration Assistant New Project type ● Assessment ○ Migration Project name SQLHack Assessment type Database Engine Source server type SQL Server Target server type Azure SQL Database 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**'	 Data Migration Assistant workshop1 Delete Assessment 1 Options · 2 Select sources · 3 Review results Select report type ☑ Check database compatibility Discover migration blocking issues and deprecated features by analyzing databases you choose in your source server to be migrated to SQL Database. ☑ Check feature parity Discover unsupported or partially-supported features and functions that your applications may rely on. Get guidance around these areas that may need some re-engineering. □ Benefit from new features (coming soon...) Discover new SQL Database features that are applicable to the databases in your source once migrated to SQL database platform. 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:** **LEGACYSQL2008 or 10.0.2.4** **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 is set to UK English not US***	 eview results Connect to a server Connect to a server and select sources Server name: LEGACYSQL2008 Authentication type: SQL Server Authentication SQL Authentication credentials Username: Demouser Password: •••••••••••••••• Connection properties □ Encrypt connection □ Trust server certificate **SQL Server permissions** To run the selected advisor(s), credentials used to connect to a source SQL Server instance must be a member of the sysadmin server role.	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.	 Add sources Connect to a server and select sources ▣ LEGACYSQL2008 ☑ TEAM01_LocalMasterDataDB ☑ TEAM01_SharedMasterDataDB ☑ TEAM01_TenantDataDb □ TEAM02_LocalMasterDataDB □ TEAM02_SharedMasterDataDB □ TEAM02_TenantDataDb □ TEAM03_LocalMasterDataDB □ TEAM03_SharedMasterDataDB □ TEAM03_TenantDataDb □ TEAM04_LocalMasterDataDB □ TEAM04_SharedMasterDataDB	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'**	 Data Migration Assistant SQLHack 1 Options ✓ 2 Select sources Add sources Remove sources Name LEGACYSQL2008 (SQL Server 2008 R2) (3) TEAM01_LocalMasterDataDB TEAM01_SharedMasterDataDB TEAM01_TenantDataDb	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.

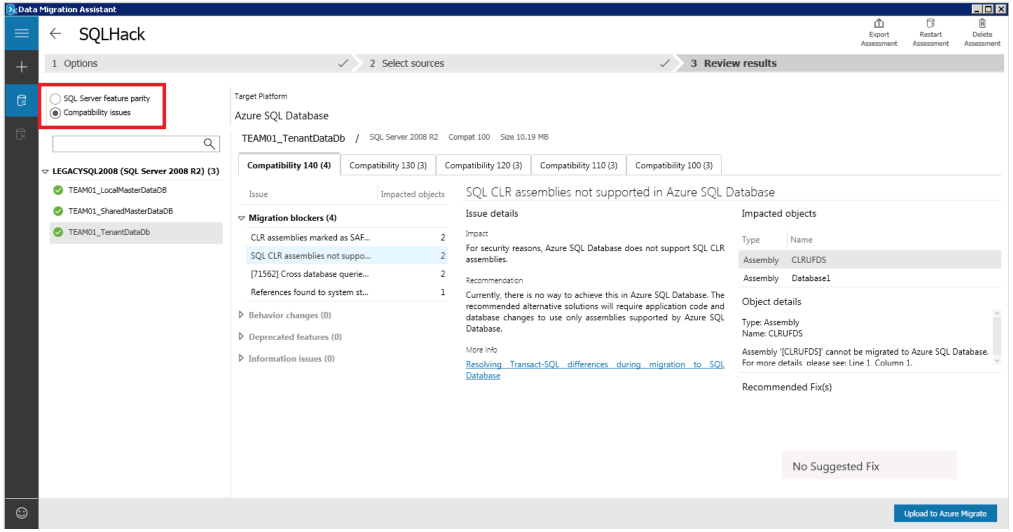
DMA will now show the results of the assessment using 2 separate reports:

‘**SQL Server feature parity**’ which is a server level report highlighting any server functionality or components (e.g. MSDTC) that the source DBs are using that isn’t fully supported on the target – in this case Azure SQL Database. In our assessment there is one ‘Unsupported feature’ reported (cross database queries).

‘**Compatibility Issues**’ which is a database level report detailing individual objects that have compatibility issues.

Select ‘**TEAMXX_TenantDataDb**’
Note the 4 ‘Migration blockers’ including CLR which the database uses.

CLR is not supported on Azure SQL DB but is supported by Azure SQL Database Managed Instance (SQLMI).



○ SQL Server feature parity
● Compatibility issues

Target Platform
Azure SQL Database

TEAM01_TenantDataDb / SQL Server 2008 R2 Compat 100 Size 10.19 MB

▽ LEGACYSQL2008 (SQL Server 2008 R2) (3)
- ✓ TEAM01_LocalMasterDataDB
- ✓ TEAM01_SharedMasterDataDB
- ✓ TEAM01_TenantDataDb

Compatibility 140 (4) | Compatibility 130 (3) | Compatibility 120 (3) | Compatibility 110 (3) | Compatibility 100 (3)

Issue	Impacted objects
▽ Migration blockers (4)	
CLR assemblies marked as SAF...	2
SQL CLR assemblies not suppo...	2
[71562] Cross database querie...	2
References found to system st...	1
▷ Behavior changes (0)	
▷ Deprecated features (0)	
▷ Information issues (0)	

SQL CLR assemblies not supported in Azure SQL Database

Issue details

Impact
For security reasons, Azure SQL Database does not support SQL CLR assemblies.

Recommendation
Currently, there is no way to achieve this in Azure SQL Database. The recommended alternative solutions will require application code and database changes to use only assemblies supported by Azure SQL Database.

More info
Resolving Transact-SQL differences during migration to SQL Database

Impacted objects

Type	Name
Assembly	CLRUFDS
Assembly	Database1

Object details

Type: Assembly
Name: CLRUFDS

Assembly '[CLRUFDS]' cannot be migrated to Azure SQL Database. For more details, please see: Line 1, Column 1.

Recommended Fix(s)

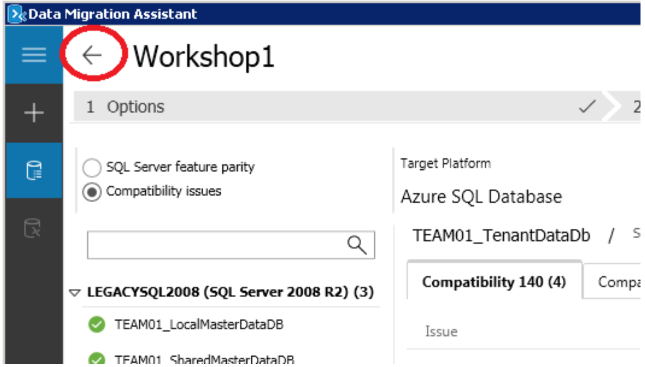
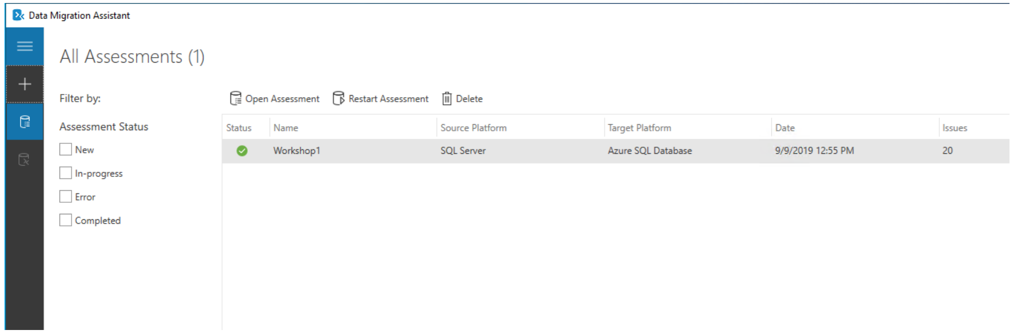
No Suggested Fix

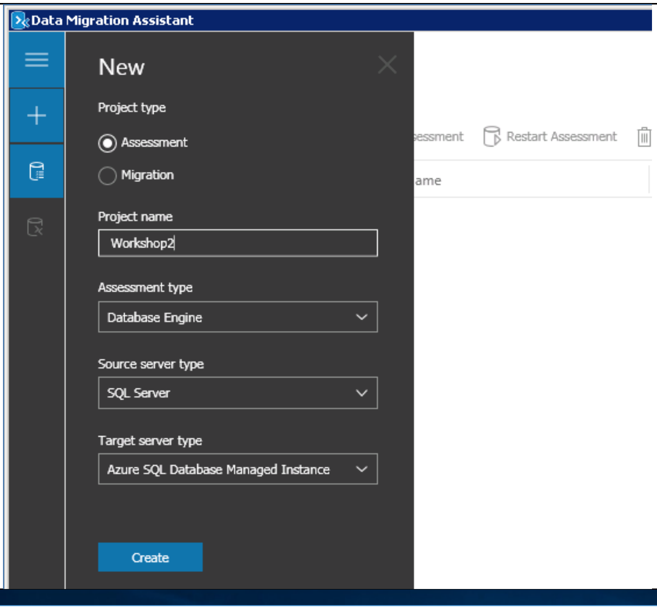
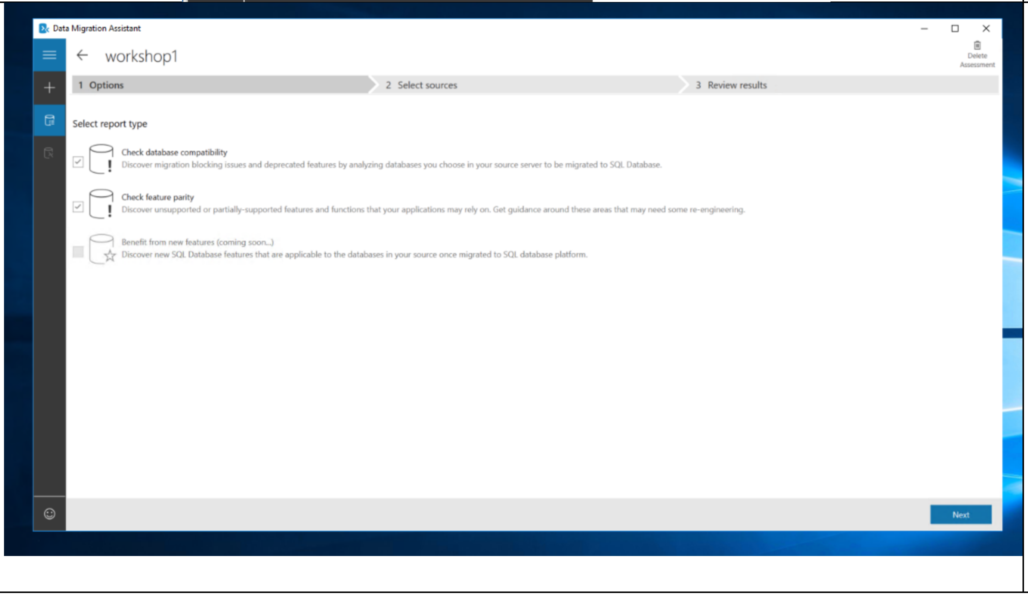
Note: Toggle the parity and compatibility issues radio button (top left) to switch between the 2 reports.

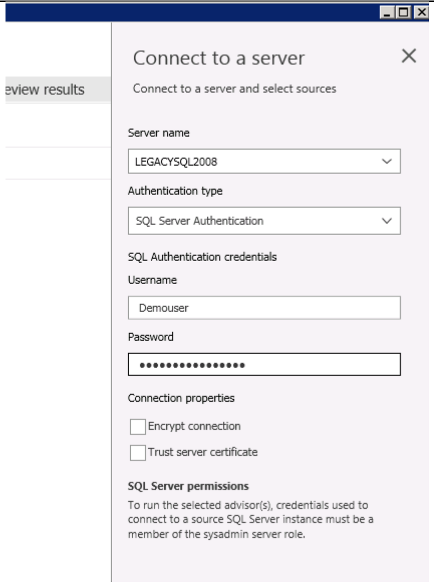
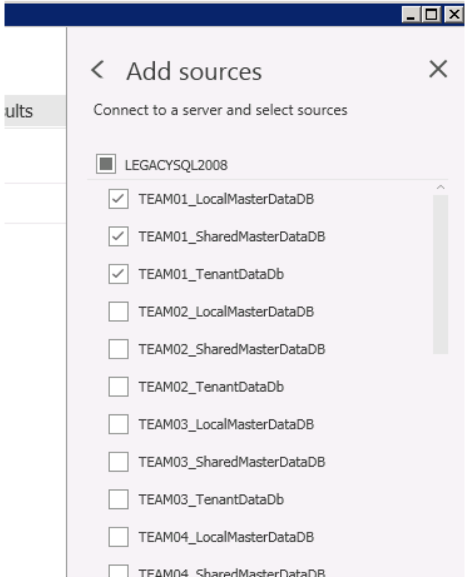
‘SQL Server feature parity’ shows what features are not supported in the target data source. Under ‘Details’ and ‘Databases’ you will find remedial action that are required and the databases impacted.

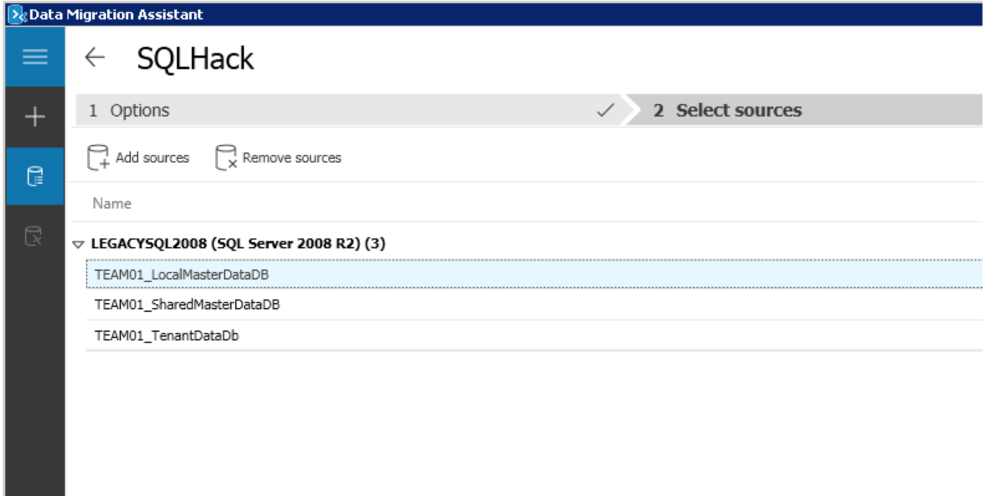
‘Compatibility Issues’ shows, over the compatibility tabs, issues that need to be addressed to permit the database(s) to run, in the chosen compatibility level (e.g. 140, 130, 120, 110).

If you have multiple databases, as with the example screenshot, you need to highlight EACH database to see the compatibility issues.

Once you’re reviewed the assessment click the back arrow to see a list of current DMA projects.	 Data Migration Assistant Workshop1 1 Options ✓ 2 ○ SQL Server feature parity ● Compatibility issues Target Platform: Azure SQL Database LEGACYSQL2008 (SQL Server 2008 R2) (3) TEAM01_LocalMasterDataDB TEAM01_SharedMasterDataDB TEAM01_TenantDataDb / S Compatibility 140 (4) | Compa Issue	
	Due to the need for CLR, we need to repeat the assessment with Azure SQL DB Managed Instance as the target.	
You should see this screenshot to the right. Select the **“+”** to create a new **assessment** project.	 Data Migration Assistant All Assessments (1) Filter by: Assessment Status □ New □ In-progress □ Error □ Completed Open Assessment | Restart Assessment | Delete Status | Name | Source Platform | Target Platform | Date | Issues ✓ | Workshop1 | SQL Server | Azure SQL Database | 9/9/2019 12:55 PM | 20	

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**'	 Data Migration Assistant New Project type ● Assessment ○ Migration Project name Workshop2 Assessment type Database Engine Source server type SQL Server Target server type Azure SQL Database Managed Instance 	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**'	 Data Migration Assistant workshop1 1 Options 2 Select sources 3 Review results Select report type ☑ Check database compatibility Discover migration blocking issues and deprecated features by analyzing databases you choose in your source server to be migrated to SQL Database. ☑ Check feature parity Discover unsupported or partially-supported features and functions that your applications may rely on. Get guidance around these areas that may need some re-engineering. □ Benefit from new features (coming soon...) Discover new SQL Database features that are applicable to the databases in your source once migrated to SQL database platform. 	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.

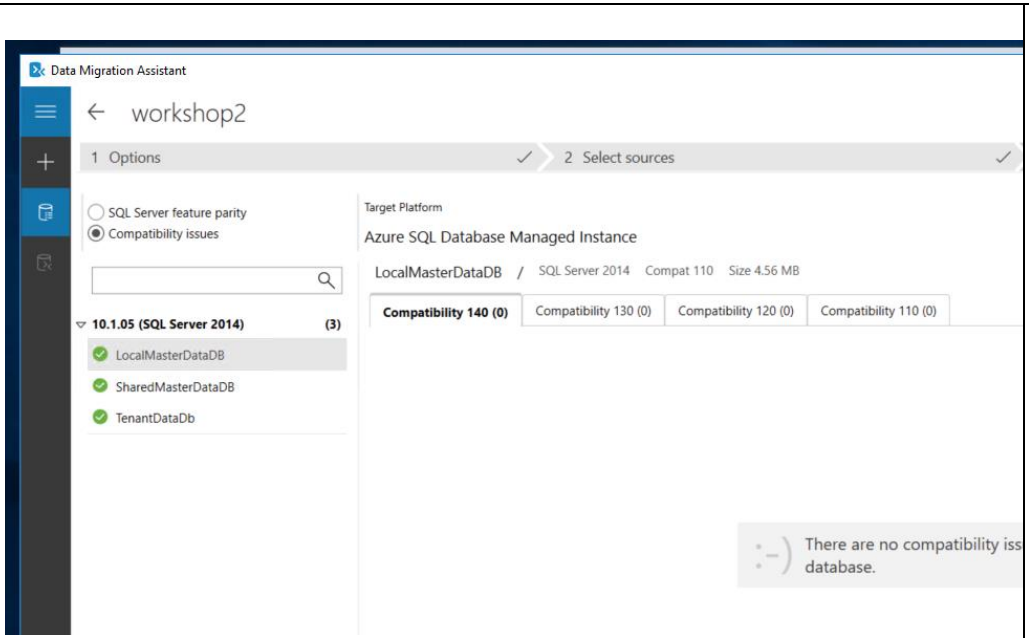
Enter the source/legacy SQL details: **Server Name:** **LEGACYSQL2008 or 10.0.2.4** **Authentication Type:** **SQL Server Authentication** **Username:** **Demouser** **Password:** **Demo@pass1234567** **Untick “Encrypt connection”** Click ‘**Connect’**	 eview results Connect to a server Connect to a server and select sources Server name LEGACYSQL2008 Authentication type SQL Server Authentication SQL Authentication credentials Username Demouser Password •••••••••••••••• Connection properties ☐ Encrypt connection ☐ Trust server certificate **SQL Server permissions** To run the selected advisor(s), credentials used to connect to a source SQL Server instance must be a member of the sysadmin server role.	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.	 ults Add sources Connect to a server and select sources ■ LEGACYSQL2008 ☑ TEAM01_LocalMasterDataDB ☑ TEAM01_SharedMasterDataDB ☑ TEAM01_TenantDataDb ☐ TEAM02_LocalMasterDataDB ☐ TEAM02_SharedMasterDataDB ☐ TEAM02_TenantDataDb ☐ TEAM03_LocalMasterDataDB ☐ TEAM03_SharedMasterDataDB ☐ TEAM03_TenantDataDb ☐ TEAM04_LocalMasterDataDB ☐ TEAM04_SharedMasterDataDB	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'**	 Data Migration Assistant SQLHack 1 Options 2 Select sources Add sources Remove sources Name LEGACYSQL2008 (SQL Server 2008 R2) (3) TEAM01_LocalMasterDataDB TEAM01_SharedMasterDataDB TEAM01_TenantDataDb	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 or it will show a single PowerShell issue for the system SQL Agent Job ‘syspolicy_purge_history’ which is not applicable to Azure SQL DB Managed Instance & can be ignored.

The ‘**Compatibility Issues**’ report should be clear for all 3 databases showing that they can be migrated to Azure SQLDB Managed Instance without changes.



Data Migration Assistant

workshop2

1 Options ✓ 2 Select sources ✓

○ SQL Server feature parity
● Compatibility issues

10.1.05 (SQL Server 2014) (3)

- LocalMasterDataDB
- SharedMasterDataDB
- TenantDataDb

Target Platform

Azure SQL Database Managed Instance

LocalMasterDataDB / SQL Server 2014 Compat 110 Size 4.56 MB

Compatibility 140 (0) | Compatibility 130 (0) | Compatibility 120 (0) | Compatibility 110 (0)

There are no compatibility iss
database.

Note: Toggle the parity and compatibility Issues radio button (top left) to see how DMA.

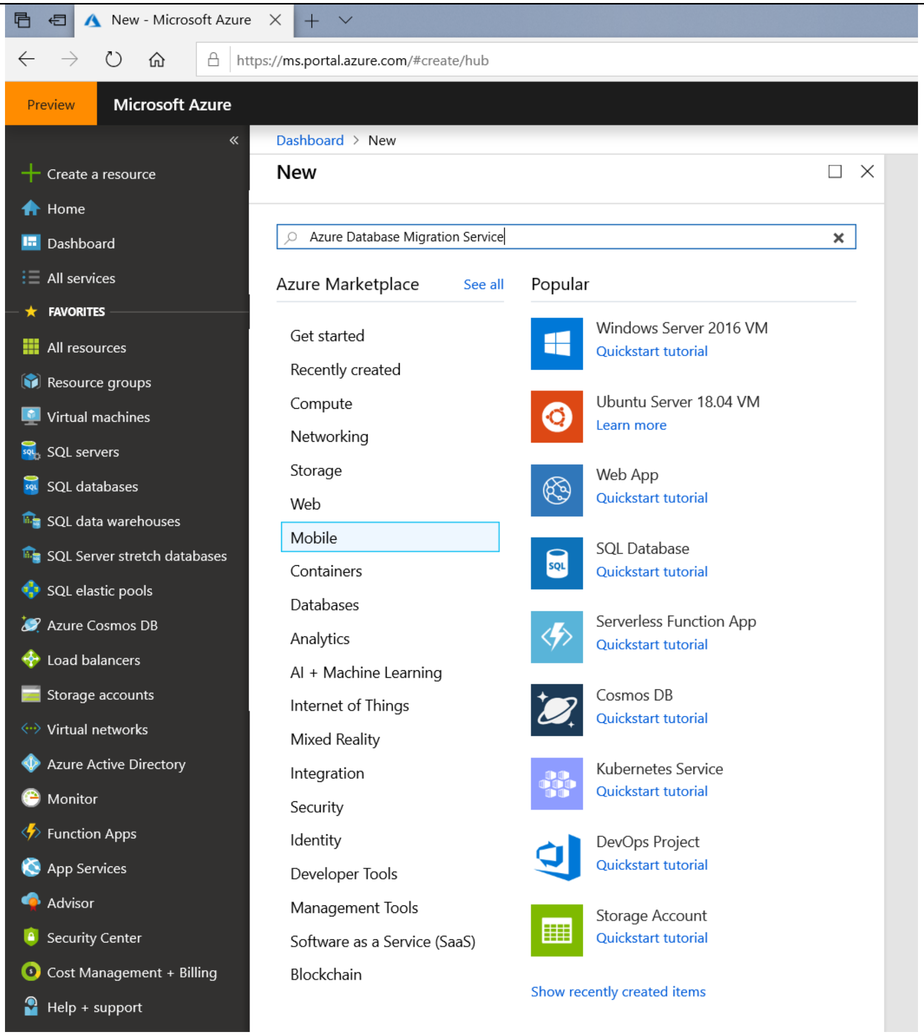
‘SQL Server feature parity’ shows what features are not supported in the target datasource. Under ‘Details’ and ‘Databases’ you will find remedial action that are required and the databases impacted.

‘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. 140, 130, 120, 110).

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. Use Azure Database Migration Service (DMS) to migrate the 3 application databases

Narrative	Screenshot	Notes
We will be using Azure Database Migration Service (DMS) to migrate the legacy SQL2008 databases to Azure. For the workshop DMS will already been provisioned as it can take 20-30mins to be deployed. If you were doing this yourself you would need to provision DMS before you could begin the migration process and would need to follow the DMS setup blades according to your organisational guidelines.		DMS is provisioned as a service which hosts & runs multiple migration Projects. Each Project is responsible for migrating one or more databases. Although a Project can migrate multiple databases each Project can only migrate databases from a single source host to a single target destination. In this lab we will use a single Project to migrate 3 databases from the same legacy SQL2008 host to Azure SQL Managed Instance. DMS can host and run different types of database migration Projects under the same instance e.g. separate Project for separate source servers. Overview of DMS: [Azure Database Migration Service]() DMS tutorials: https://docs.microsoft.com/en-us/azure/dms/

https://ms.portal.azure.com/#create/hub

Azure Database Migration Service

For this workshop:
On your Win10 VM open Edge browser and got to:

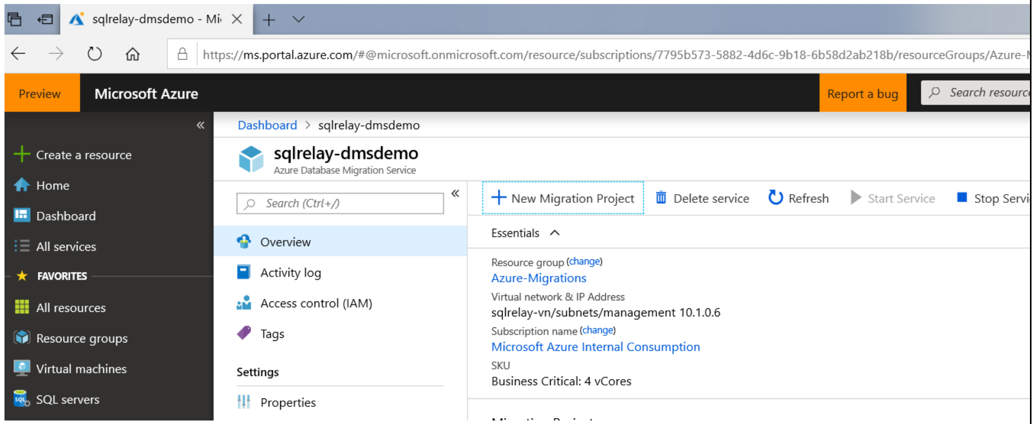
HTTPS://portal.azure.com

When prompted for login details use the Azure credentials you used previously from the Excel workbook:

Username:
SQLHACK_TEAMXXxxx@OTAPRD672ops.onmicrosoft.com
Password:
<Azure portal password>

In the Azure portal, open the **SQLHACK-SHARED Resource Group** and locate the **Azure Database Migration Service** and open it.

On the DMS Overview blade click '**New Migration Project'**



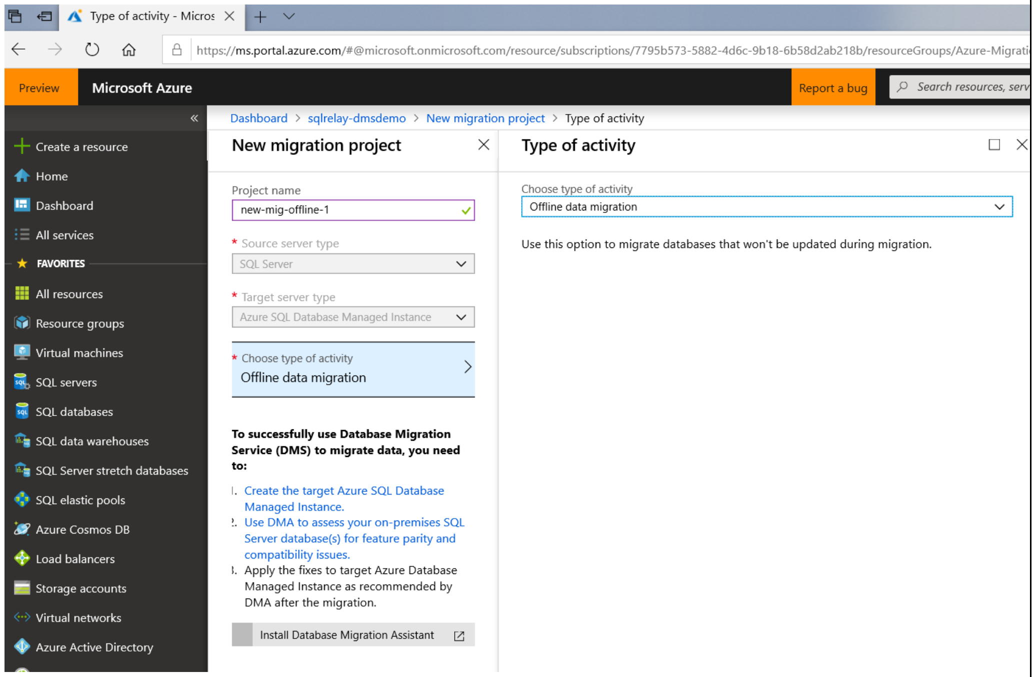
https://ms.portal.azure.com/#@microsoft.onmicrosoft.com/resource/subscriptions/7795b573-5882-4d6c-9b18-6b58d2ab218b/resourceGroups/Azure-

Dashboard > sqlrelay-dmsdemo

sqlrelay-dmsdemo
Azure Database Migration Service

Essentials

Resource group (change)
Azure-Migrations
Virtual network & IP Address
sqlrelay-vn/subnets/management 10.1.0.6
Subscription name (change)
Microsoft Azure Internal Consumption
SKU
Business Critical: 4 vCores

On the “New migration project” blade use the following settings: **Project name:** **TEAMXX_migration** *(replace XX with your team number)* **Source server type:** **SQL Server** **Target Server type:** **Azure SQL Database Managed Instance** **Choose type of activity:** **Offline data migration** Click ‘**Save**’ Select ‘**Create and run activity**’	 Type of activity - Micros https://ms.portal.azure.com/#@microsoft.onmicrosoft.com/resource/subscriptions/7795b573-5882-4d6c-9b18-6b58d2ab218b/resourceGroups/Azure-Migrati Preview Microsoft Azure Report a bug Search resources, serv Create a resource Home Dashboard All services FAVORITES All resources Resource groups Virtual machines SQL servers SQL databases SQL data warehouses SQL Server stretch databases SQL elastic pools Azure Cosmos DB Load balancers Storage accounts Virtual networks Azure Active Directory Dashboard > sqlrelay-dmsdemo > New migration project > Type of activity New migration project Project name: new-mig-offline-1 * Source server type: SQL Server * Target server type: Azure SQL Database Managed Instance * Choose type of activity: Offline data migration **To successfully use Database Migration Service (DMS) to migrate data, you need to:** 1. Create the target Azure SQL Database Managed Instance. 2. Use DMA to assess your on-premises SQL Server database(s) for feature parity and compatibility issues. 3. Apply the fixes to target Azure Database Managed Instance as recommended by DMA after the migration. Install Database Migration Assistant Type of activity Choose type of activity: Offline data migration Use this option to migrate databases that won't be updated during migration.	DMS can perform two types of database migrations: - **Offline** - **Online** **Offline** migrations use backup files. The backups can be provided to DMS or DMS can create the backup as part of a project. Whilst the simplest to perform, taking the backup, moving it to Azure and restoring it can cause significant downtime. **Online** migrations use a replication or log shipping approach to keep the source & target in sync. Whilst more complex it significantly reduces database downtime.

DMS will now launch the migration configuration blades. Use the following values for each of the configuration steps:

STEP 1: Select Source

This uses the source database host VM LEGACYSQL2008 details from the "Lab and parameters" doc.

Source SQL Server:
LEGACYSQL2008 or 10.0.2.4

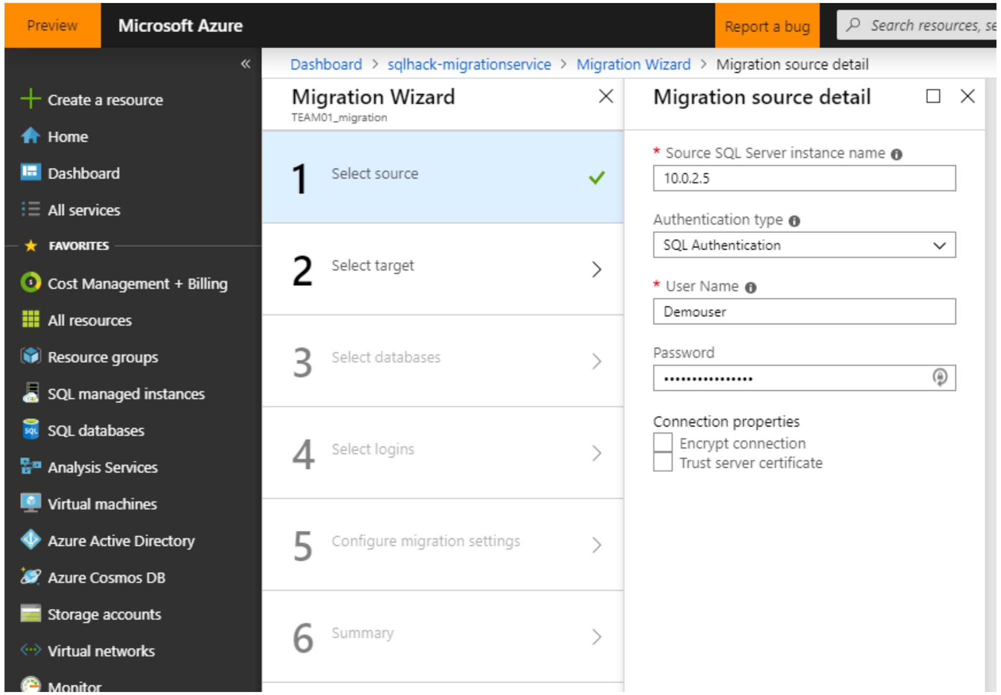
Authentication Type:
SQL Authentication

User Name:
Demouser

Password
Demo@pass1234567

For this lab only uncheck both "Connection Properties" options as per the screenshot.

Click '**Save**'.
This will perform a connection test.



Dashboard > sqlhack-migrationservice > Migration Wizard > Migration source detail

Migration Wizard
TEAM01_migration

1 Select source ✓
2 Select target
3 Select databases
4 Select logins
5 Configure migration settings
6 Summary

Migration source detail

* Source SQL Server instance name: 10.0.2.5

Authentication type: SQL Authentication

* User Name: Demouser

Password: ••••••••••••••••

Connection properties
□ Encrypt connection
□ Trust server certificate

The account that DMS uses to connect to the source instance must be a member of sysadmin.

STEP 2: Select Target

This uses the target Azure SQL Managed Instance details from the “Lab and parameters” doc.

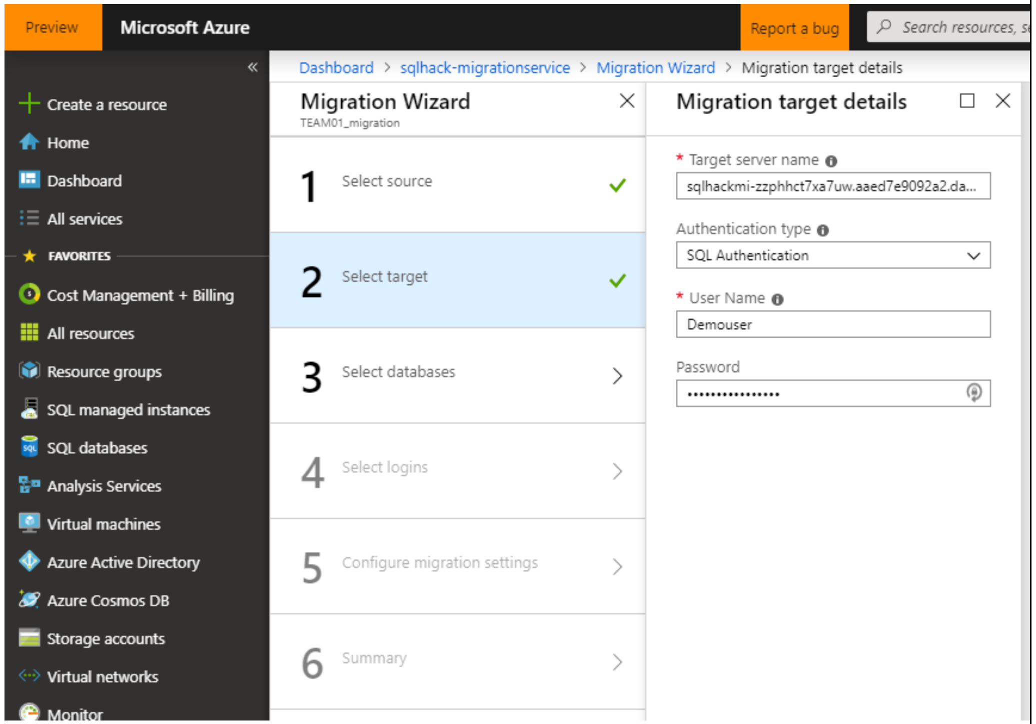
Target SQL Server:
(Use the Fully Qualified Domain Name for the SQL Managed Instance which can be found in C:_SQLHACK_\LABS\01-Data_Migration\ManagedInstanceFDQN.txt

Authentication Type:
SQL Authentication

User Name:
Demouser

Password:
Demo@pass1234567

Click ‘**Save**’.
This will perform a connection test.



* Target server name: sqlhackmi-zzphhct7xa7uw.aaed7e9092a2.da...

Authentication type: SQL Authentication

* User Name: Demouser

Password: ••••••••••••••••

STEP 3: Select Databases

The application has 3 databases supporting it. Select the 3 databases for you team.

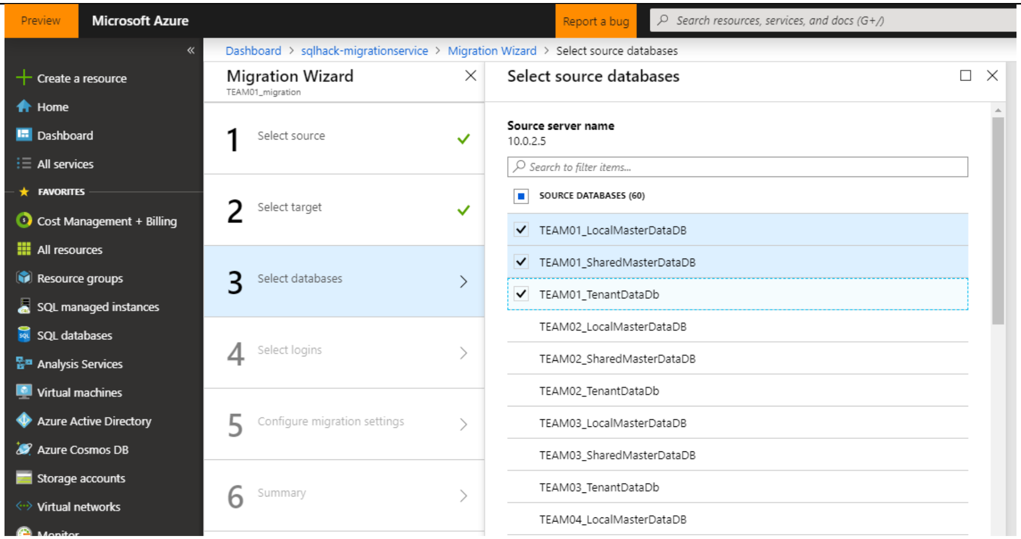
TEAMXX_LocalMasterDataDb

TEAMXX_SharedMasterDb

TEAMXX_TenantDataDb

(replace XX with your team number)

Select '**Save**'



Dashboard > sqlhack-migrationservice > Migration Wizard > Select source databases

Migration Wizard
TEAM01_migration

1 Select source ✓
2 Select target ✓
3 Select databases >
4 Select logins >
5 Configure migration settings >
6 Summary >

Select source databases

Source server name
10.0.2.5

Search to filter items...

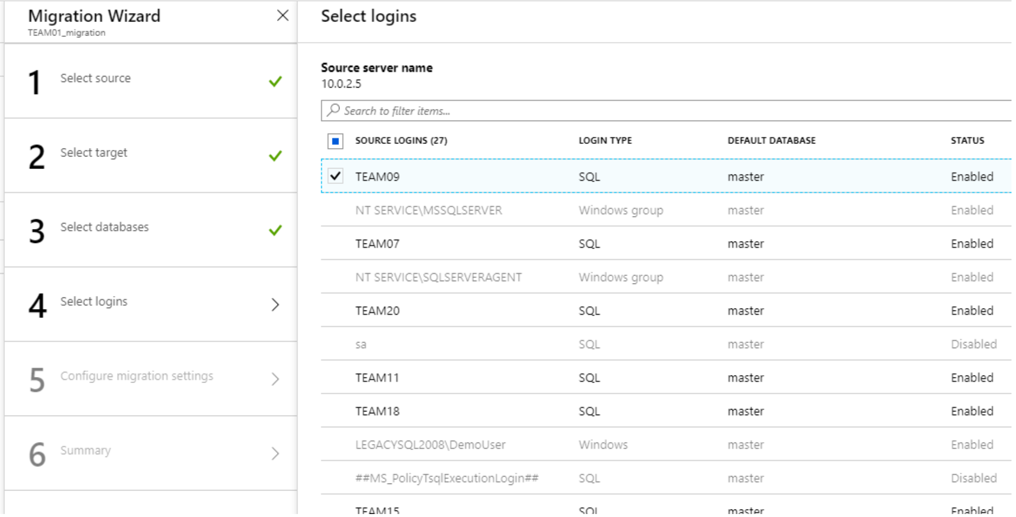
	SOURCE DATABASES (60)
☑	TEAM01_LocalMasterDataDB
☑	TEAM01_SharedMasterDataDB
☑	TEAM01_TenantDataDb
	TEAM02_LocalMasterDataDB
	TEAM02_SharedMasterDataDB
	TEAM02_TenantDataDb
	TEAM03_LocalMasterDataDB
	TEAM03_SharedMasterDataDB
	TEAM03_TenantDataDb
	TEAM04_LocalMasterDataDB

STEP 4: Select Logins

As with a traditional on-premise migrations the SQL Server level logins must be migrated alongside the database. Select the database logins, from the list, that are required for the application.

Select ***only*** your '**TEAMXX**' login.

Select '**Save**'



Migration Wizard
TEAM01_migration

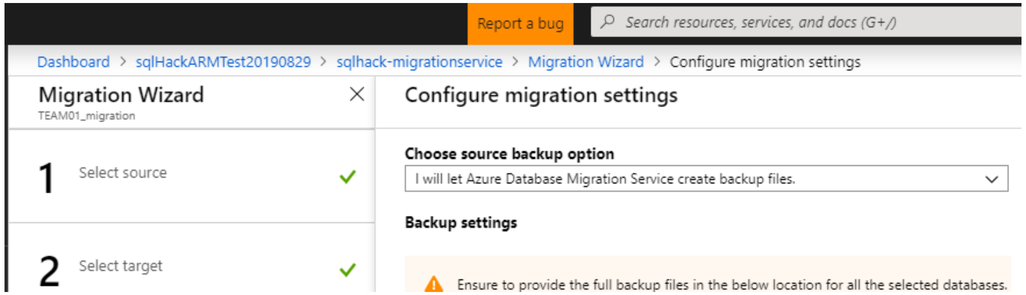
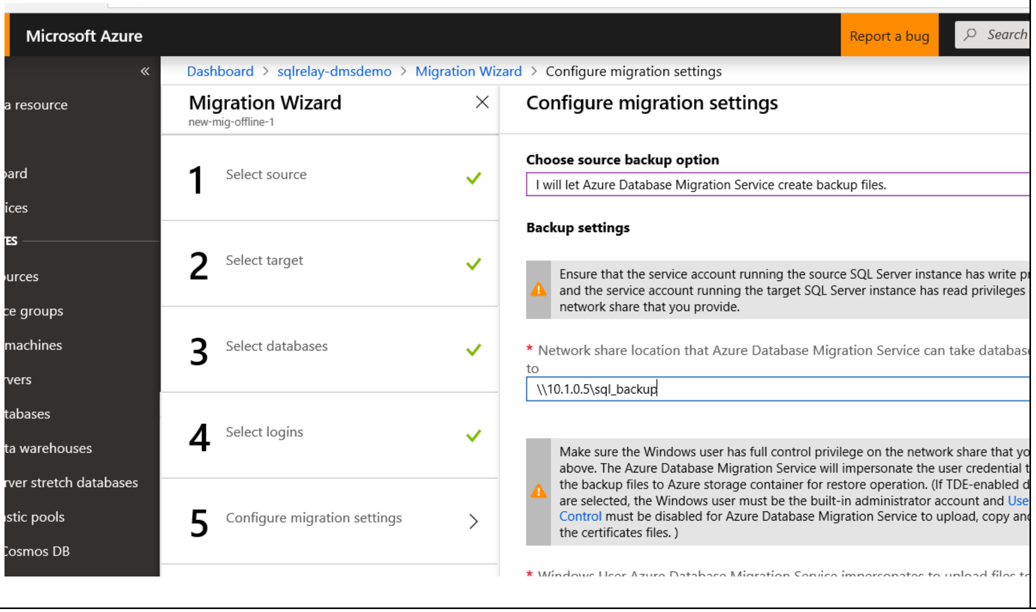
1 Select source ✓
2 Select target ✓
3 Select databases ✓
4 Select logins >
5 Configure migration settings >
6 Summary >

Select logins

Source server name
10.0.2.5

Search to filter items...

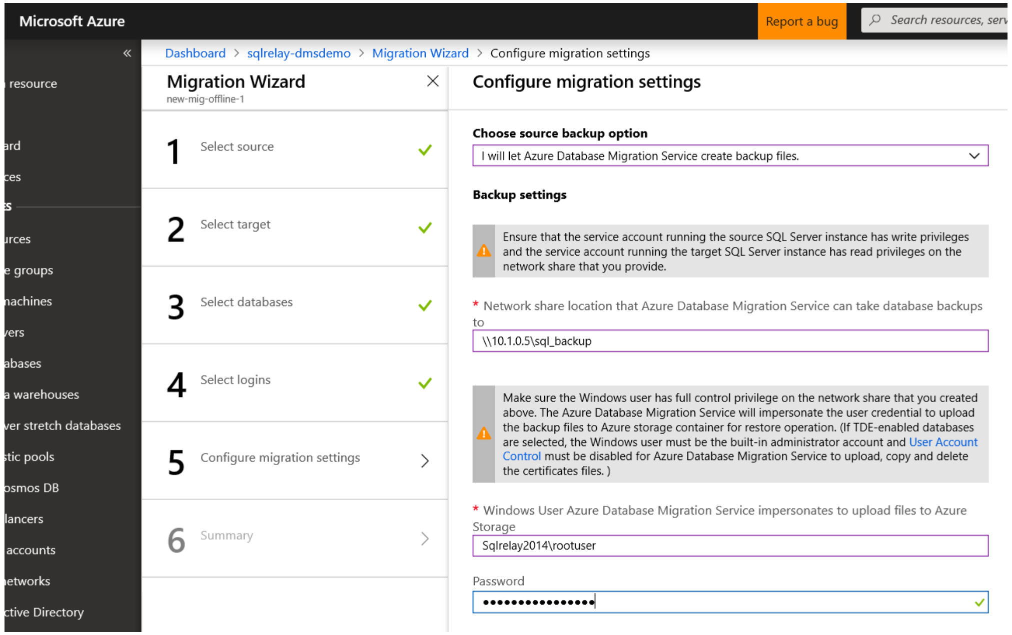
	SOURCE LOGINS (27)	LOGIN TYPE	DEFAULT DATABASE	STATUS
☑	TEAM09	SQL	master	Enabled
	NT SERVICE\MSSQLSERVER	Windows group	master	Enabled
	TEAM07	SQL	master	Enabled
	NT SERVICE\SQLSERVERAGENT	Windows group	master	Enabled
	TEAM20	SQL	master	Enabled
	sa	SQL	master	Disabled
	TEAM11	SQL	master	Enabled
	TEAM18	SQL	master	Enabled
	LEGACYSQL2008\DemoUser	Windows	master	Enabled
	##MS_PolicyTsqlExecutionLogin##	SQL	master	Disabled
	TEAM15	SQL	master	Enabled

Step 5a: Configure migration Settings ***(Source Backup Option)*** We are running an offline migration which will use backups of the DBs to be migrated. We want DMS to perform the backups, so select this option from the “Choose source backup option” (as shown).	 Dashboard > sqlHackARMTest20190829 > sqlhack-migrationservice > Migration Wizard > Configure migration settings Migration Wizard TEAM01_migration 1 Select source ✓ 2 Select target ✓ Configure migration settings **Choose source backup option** I will let Azure Database Migration Service create backup files. **Backup settings** ⚠ Ensure to provide the full backup files in the below location for all the selected databases.	DMS can use backup files provided or take the backups as part of the migration activity.
Step 5b: Configure migration Settings ***(backup location)*** We can now enter the Windows share that the source server will write the database backups to. **Network Share:** **\\10.0.2.4\FILESHARE** **or** **\\LEGACYSQL2008\FILESHARE**	 Microsoft Azure Report a bug Search a resource ard ices ES urces e groups machines vers tabases ta warehouses ver stretch databases stic pools Cosmos DB Dashboard > sqlrelay-dmsdemo > Migration Wizard > Configure migration settings Migration Wizard new-mig-offline-1 1 Select source ✓ 2 Select target ✓ 3 Select databases ✓ 4 Select logins ✓ 5 Configure migration settings > Configure migration settings **Choose source backup option** I will let Azure Database Migration Service create backup files. **Backup settings** ⚠ Ensure that the service account running the source SQL Server instance has write pr and the service account running the target SQL Server instance has read privileges network share that you provide. * Network share location that Azure Database Migration Service can take databas to \\10.1.0.5\sql_backup ⚠ Make sure the Windows user has full control privilege on the network share that yo above. The Azure Database Migration Service will impersonate the user credential t the backup files to Azure storage container for restore operation. (If TDE-enabled d are selected, the Windows user must be the built-in administrator account and Use Control must be disabled for Azure Database Migration Service to upload, copy and the certificates files.) * Windows User Azure Database Migration Service impersonates to upload files to	

Step 5c: Configure migration Settings (Windows User for DMS)

We now provide the username and password of the windows account that will permit the DMS service to run the backups on the source host and save them to the share on the legacy server.

Storage (Username):
LEGACYSQL2008\Demouser
Password:
Demo@pass1234567



Dashboard > sqlrelay-dmsdemo > Migration Wizard > Configure migration settings

Migration Wizard
new-mig-offline-1

1 Select source ✓
2 Select target ✓
3 Select databases ✓
4 Select logins ✓
5 Configure migration settings >
6 Summary >

Configure migration settings

Choose source backup option
I will let Azure Database Migration Service create backup files.

Backup settings

Ensure that the service account running the source SQL Server instance has write privileges and the service account running the target SQL Server instance has read privileges on the network share that you provide.

* Network share location that Azure Database Migration Service can take database backups to
\\10.1.0.5\sql_backup

Make sure the Windows user has full control privilege on the network share that you created above. The Azure Database Migration Service will impersonate the user credential to upload the backup files to Azure storage container for restore operation. (If TDE-enabled databases are selected, the Windows user must be the built-in administrator account and User Account Control must be disabled for Azure Database Migration Service to upload, copy and delete the certificates files.)

* Windows User Azure Database Migration Service impersonates to upload files to Azure Storage
Sqlrelay2014\rootuser

Password
••••••••••••••••

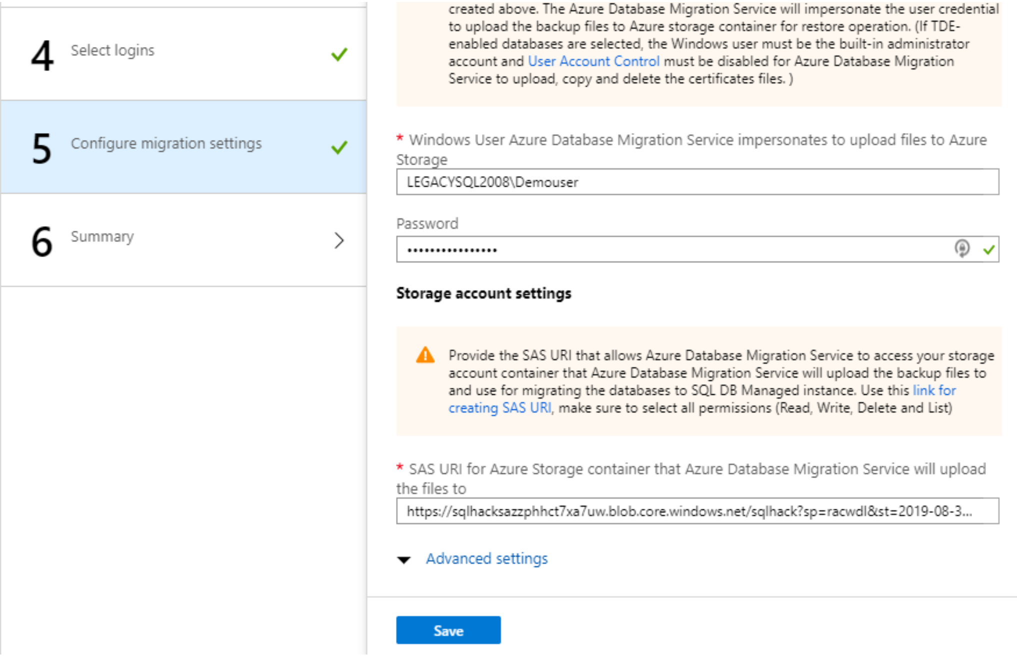
Step 5d: Configure migration Settings *(Azure storage account settings)*

DMS is an Azure Service. We have to provide the Shared Access Signature token (or "SAS URI" for short) to permit DMS to upload the backup files, from the share on the LEGACYSQL2008 host to Azure blob storage where the SQL Managed Instance can access them.

The SAS URI is both the URL of a contain (folder) in Azure Blob Storage and the key to access it.

The SAS URI can be found in: **C:_SQLHACK_\LABS\01-Data Migration\SASKey.txt**

Enter the SAS URI key and click '**Save**'.
This will perform a connection test and if successful will display the **Summary** blade.



created above. The Azure Database Migration Service will impersonate the user credential to upload the backup files to Azure storage container for restore operation. (If TDE-enabled databases are selected, the Windows user must be the built-in administrator account and User Account Control must be disabled for Azure Database Migration Service to upload, copy and delete the certificates files.)

* Windows User Azure Database Migration Service impersonates to upload files to Azure Storage: LEGACYSQL2008\Demouser

Password: ················ ✓

Storage account settings

Provide the SAS URI that allows Azure Database Migration Service to access your storage account container that Azure Database Migration Service will upload the backup files to and use for migrating the databases to SQL DB Managed instance. Use this link for creating SAS URI, make sure to select all permissions (Read, Write, Delete and List)

* SAS URI for Azure Storage container that Azure Database Migration Service will upload the files to: https://sqlhacksazzphhct7xa7uw.blob.core.windows.net/sqlhack?sp=racwdl&st=2019-08-3...

Once DMS has taken backups of the databases to be migrated it needs to move these backups to Azure storage. This is so the target SQL Managed Instance can access them to restore them.

STEP 5e: Configure migration Settings *(Summary)*

DMS displays the migration configuration settings.

Now we need to use these settings to actually perform a migration. To do this we create an “Activity”.

On the **Summary** settings use the following values:

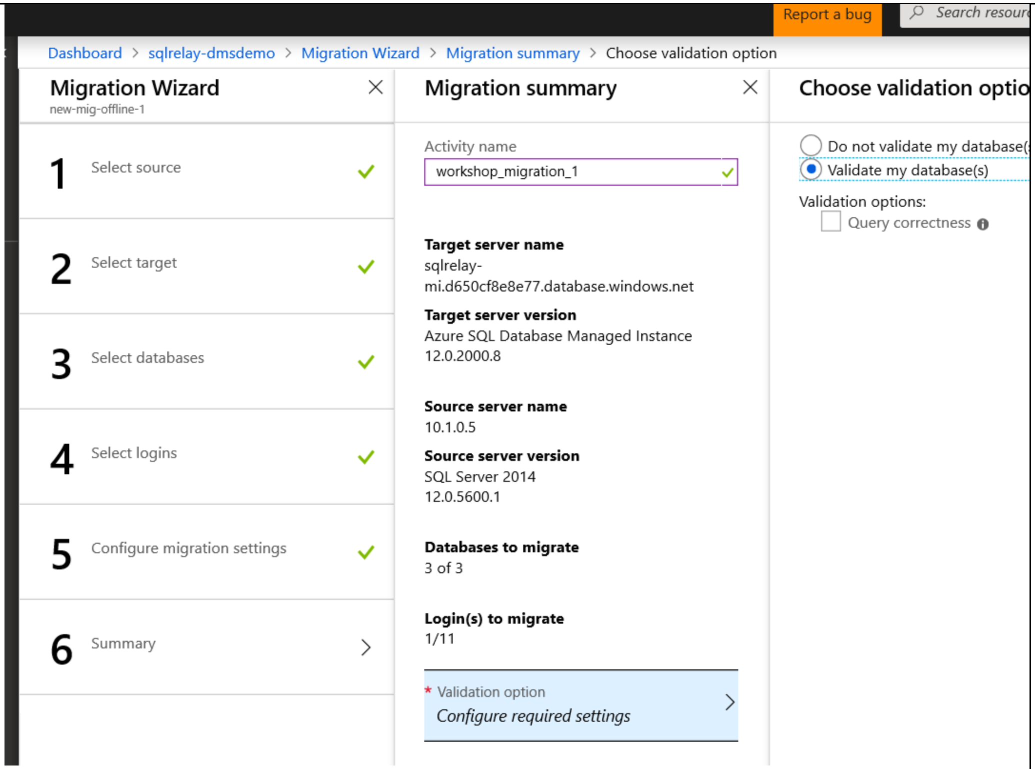
Activity Name

workshop_migration_XX *(replace XX with your team number)*

Validation option:

Validate my database(s)’

Select ‘**Save**’



Report a bug | Search resour

Dashboard > sqlrelay-dmsdemo > Migration Wizard > Migration summary > Choose validation option

Migration Wizard
new-mig-offline-1

1 Select source ✓

2 Select target ✓

3 Select databases ✓

4 Select logins ✓

5 Configure migration settings ✓

6 Summary >

Migration summary

Activity name: workshop_migration_1 ✓

Target server name
sqlrelay-mi.d650cf8e8e77.database.windows.net

Target server version
Azure SQL Database Managed Instance
12.0.2000.8

Source server name
10.1.0.5

Source server version
SQL Server 2014
12.0.5600.1

Databases to migrate
3 of 3

Login(s) to migrate
1/11

* Validation option
Configure required settings >

Choose validation optio

○ Do not validate my database(

● Validate my database(s)

Validation options:

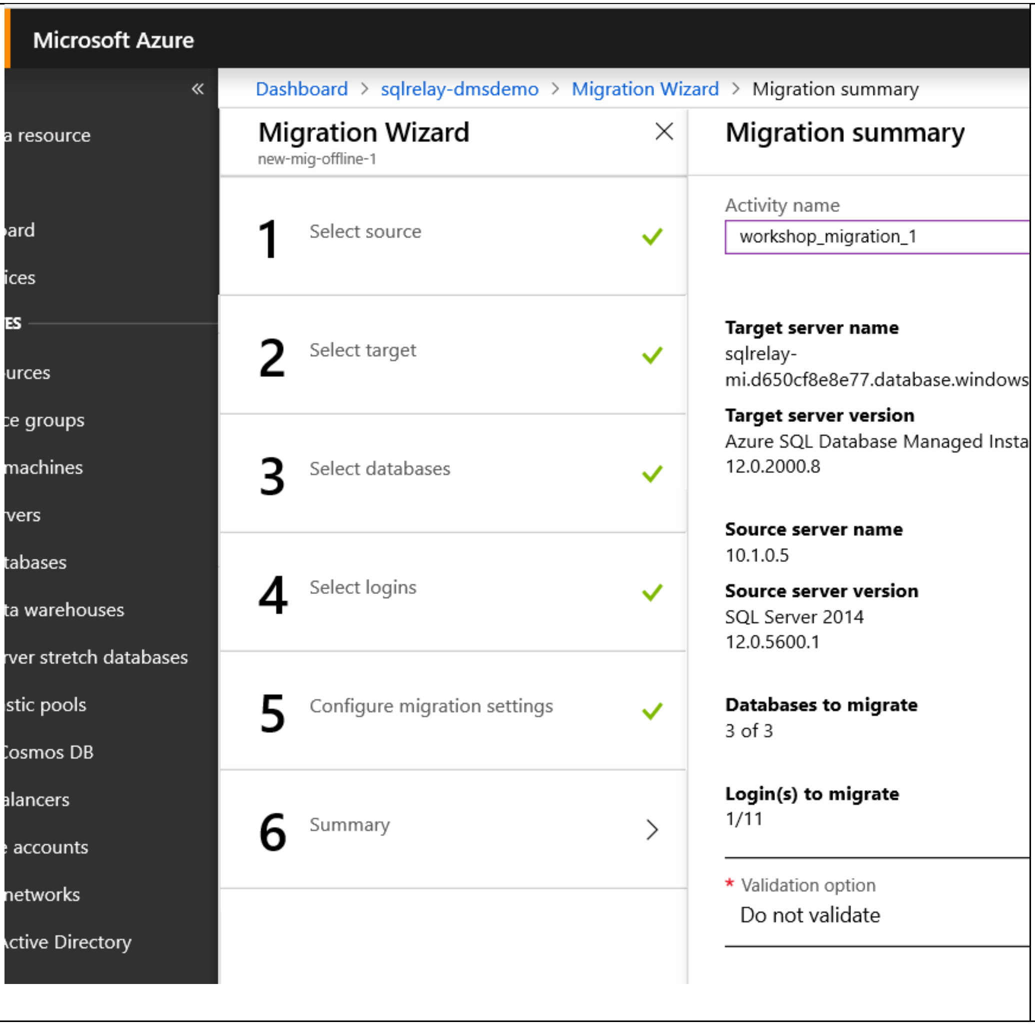
□ Query correctness

Validate my databases option

Selecting this validation option forces DMS to do the following tasks:

1. Takes top 100 resource intense queries and re-runs them against the target and reports the success/failure rate
2. Table checksum on all rows and report any differences

This can be an intensive process so best test it with a non-production migration to see how much extra time it adds to the migration.

STEP 5f: Migration Summary ***(run the migration)*** The migration activity is now configured and saved and is ready to run. Select '**Run migration'**		

Migration Wizard
new-mig-offline-1

1 Select source ✓
2 Select target ✓
3 Select databases ✓
4 Select logins ✓
5 Configure migration settings ✓
6 Summary >

Migration summary

Activity name: workshop_migration_1

Target server name
sqlrelay-mi.d650cf8e8e77.database.windows

Target server version
Azure SQL Database Managed Insta
12.0.2000.8

Source server name
10.1.0.5

Source server version
SQL Server 2014
12.0.5600.1

Databases to migrate
3 of 3

Login(s) to migrate
1/11

* Validation option
Do not validate

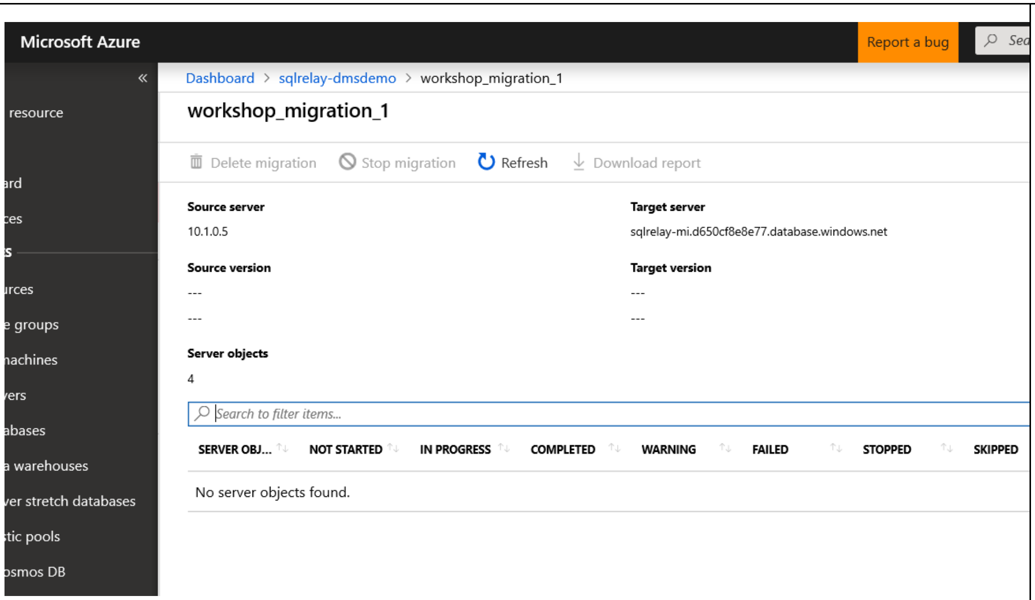
DMS will now run the migration activity.

Initially this screen will be displayed.

Select ‘**Refresh**’ to monitor the progress of your migration.

Notice the database counts under the following columns as you keep pressing ‘**Refresh**’:

“IN PROGRESS”
“COMPLETED”
“FAILED”



workshop_migration_1

Source server
10.1.0.5

Target server
sqlrelay-mi.d650cf8e8e77.database.windows.net

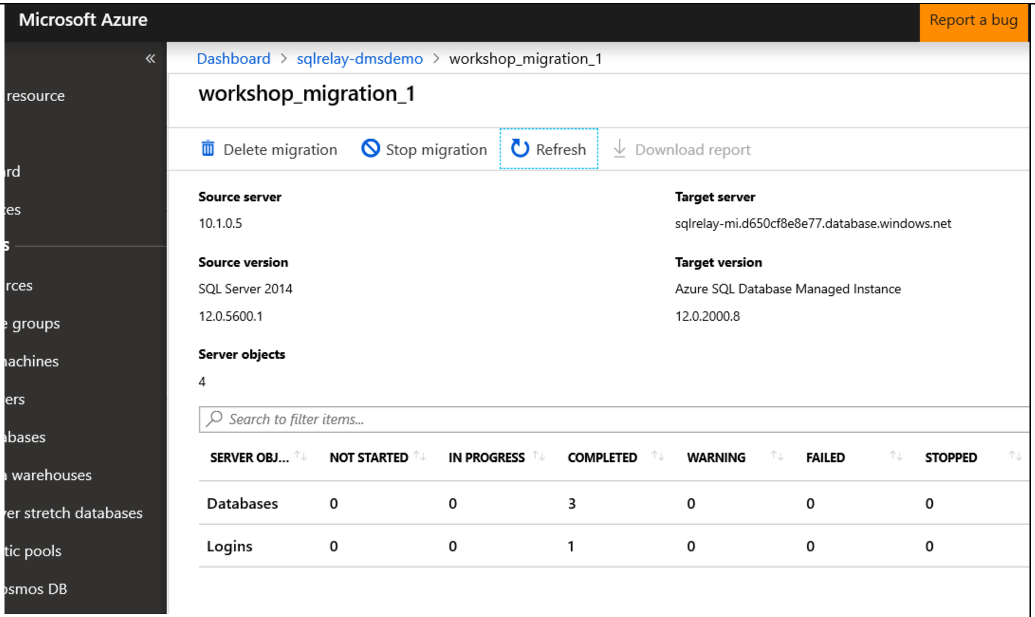
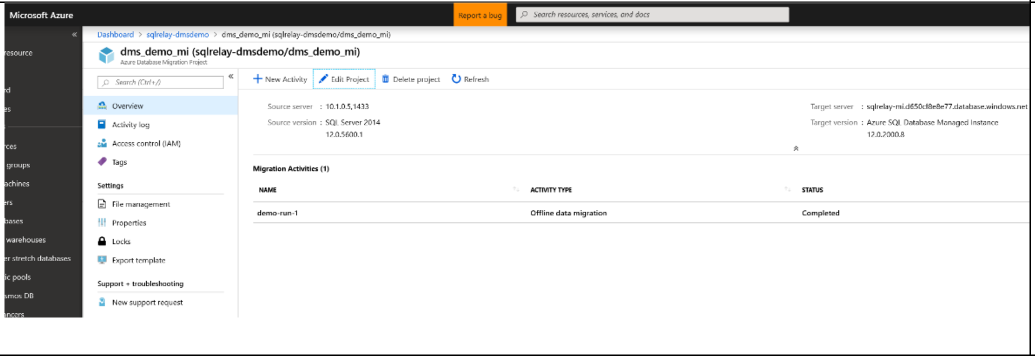
Source version

Target version

Server objects
4

Search to filter items...

SERVER OBJ...	NOT STARTED	IN PROGRESS	COMPLETED	WARNING	FAILED	STOPPED	SKIPPED
No server objects found.							

Under "COMPLETED", when the number of databases says "3" the migration activity has completed.		If there any warnings, errors or skipped databases they will have a database count under the corresponding headings in the status page. It is assumed, for the workshop, that all three databases have migrated successfully.
Close the migration activity. On the migration project blade notice that your migration activity is displayed.		DMS keeps a history of activity runs for migration projects. A migration activity can be edited and ran again.

Dashboard > sqlrelay-dmsdemo > workshop_migration_1

workshop_migration_1

Source server
10.1.0.5

Target server
sqlrelay-mi.d650cf8e8e77.database.windows.net

Source version
SQL Server 2014
12.0.5600.1

Target version
Azure SQL Database Managed Instance
12.0.2000.8

Server objects
4

SERVER OBJ...	NOT STARTED	IN PROGRESS	COMPLETED	WARNING	FAILED	STOPPED
Databases	0	0	3	0	0	0
Logins	0	0	1	0	0	0

Dashboard > sqlrelay-dmsdemo > dms_demo_mi (sqlrelay-dmsdemo/dms_demo_mi)

dms_demo_mi (sqlrelay-dmsdemo/dms_demo_mi)
Azure Database Migration Project

Source server : 10.1.0.5,1433
Source version : SQL Server 2014
12.0.5600.1

Target server : sqlrelay-mi.d650cf8e8e77.database.windows.net
Target version : Azure SQL Database Managed Instance
12.0.2000.8

Migration Activities (1)

NAME	ACTIVITY TYPE	STATUS
demo-run-1	Offline data migration	Completed

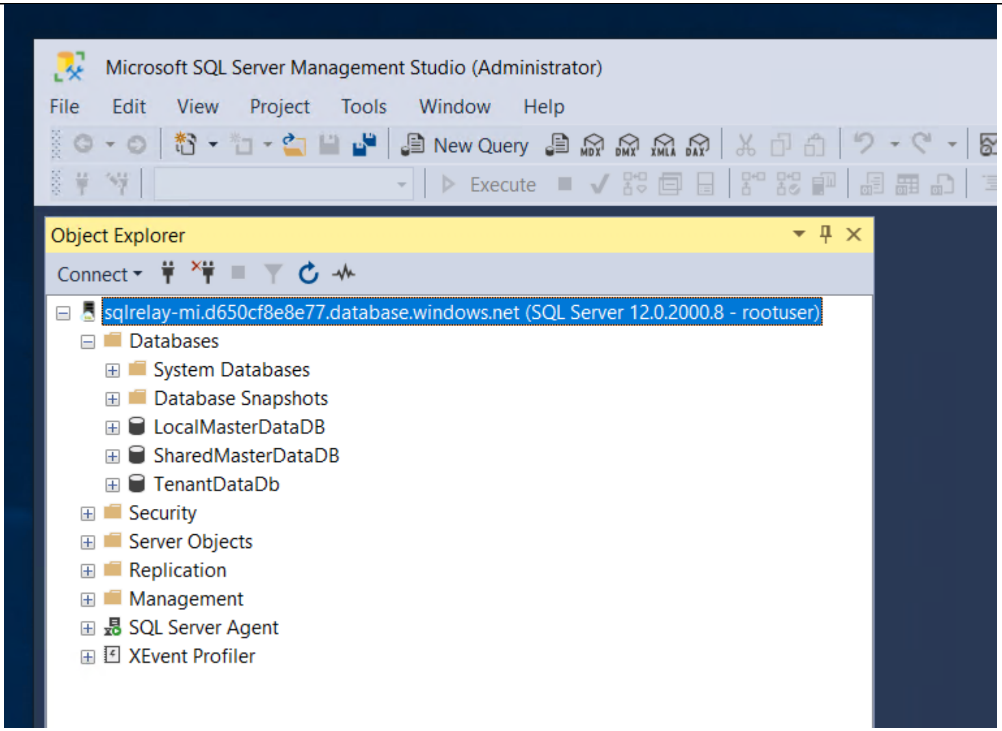
4. Confirm application databases have been migrated to Azure SQL Managed Instance

On your Win10 VM open SQL Management Studio and connect to the target Azure SQL Database Managed Instance using these details:

Server:
Use the Fully Qualified Domain Name for the SQL Managed Instance which can be found in:
C:_SQLHACK_\LABS\01-Data_Migration\ManagedInstanceFDQN.txt
SQL Authentication
Username:
TEAMXX
Password:
TEAMXX

Open the 'Databases' folder and verify the three databases have been migrated and are online.

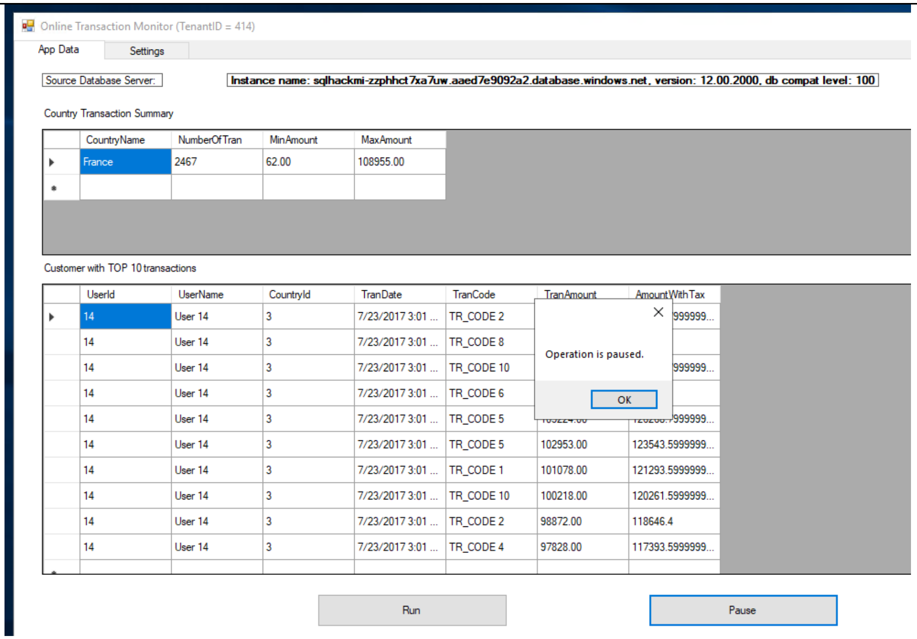
TEAMXX_LocalMasterDataDb
TEAMXX_SharedMasterDb
TEAMXX_TenantDataDb



SSMS should be installed, for this workshop, on the Windows VM.

5. Connect ‘Online Transaction Monitor’ App to Azure SQL DB Managed Instance

Now that we have migrated the databases to Azure we need to restart the application to use the new database.

Narrative	Screenshot	Notes
On your team Win10 management VM run the SimpleTranReportApp application. Note: You will likely already have the app loaded from the earlier stage in this workshop. IF it is still running simulated transactions, click ‘**Pause**’		

Online Transaction Monitor (TenantID = 414)

App Data | Settings

Source Database Server: Instance name: sqlhackmi-zzphhct7xa7uw.aaed7e9092a2.database.windows.net, version: 12.00.2000, db compat level: 100

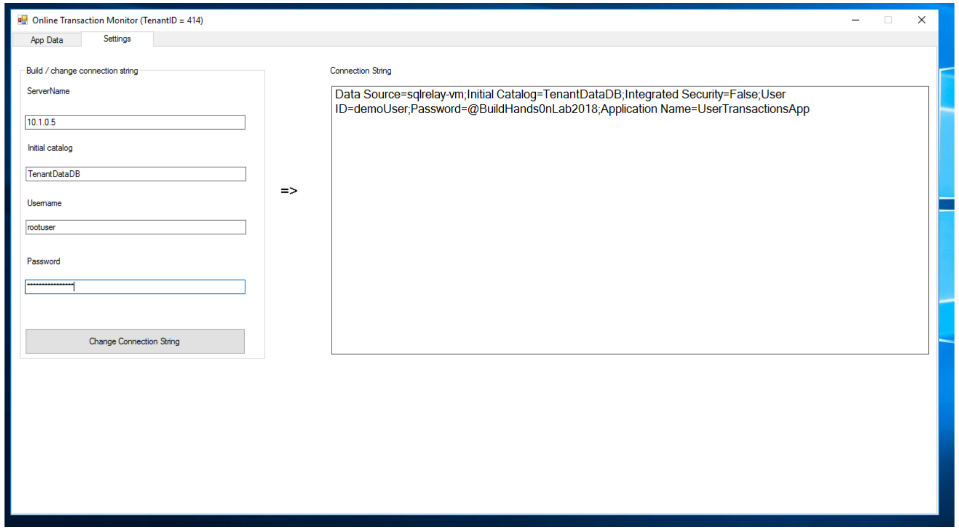
Country Transaction Summary

CountryName	NumberOfTran	MinAmount	MaxAmount
France	2467	62.00	108955.00

Customer with TOP 10 transactions

UserId	UserName	CountryId	TranDate	TranCode	TranAmount	AmountWithTax
14	User 14	3	7/23/2017 3:01 ...	TR_CODE 2		999999...
14	User 14	3	7/23/2017 3:01 ...	TR_CODE 8		
14	User 14	3	7/23/2017 3:01 ...	TR_CODE 10		999999...
14	User 14	3	7/23/2017 3:01 ...	TR_CODE 6		
14	User 14	3	7/23/2017 3:01 ...	TR_CODE 5	[illegible]	[illegible]999999...
14	User 14	3	7/23/2017 3:01 ...	TR_CODE 5	102953.00	123543.5999999...
14	User 14	3	7/23/2017 3:01 ...	TR_CODE 1	101078.00	121293.5999999...
14	User 14	3	7/23/2017 3:01 ...	TR_CODE 10	100218.00	120261.5999999...
14	User 14	3	7/23/2017 3:01 ...	TR_CODE 2	98872.00	118646.4
14	User 14	3	7/23/2017 3:01 ...	TR_CODE 4	97828.00	117393.5999999...

Operation is paused.

Reconfigure the applications connection string so it's connects to the newly migrated databases on the SQL Managed Instance. Once running, select the '**Settings**' tab Enter the following parameters into the fields identified: **ServerName:** *Use the FQDN for the SQL Managed Instance which can be found in:* **C:_SQLHACK_\LABS\01-Data_Migration\ManagedInstanceFDQN.txt** **Initial Catalog:** **TEAMXX_TenantDataDb** **UserName:** **TEAMXX** **Password:** **TEAMXX** Click '**Change Connection String**' to apply these new settings.	 Online Transaction Monitor (TenantID = 414) Build / change connection string ServerName: 10.1.0.5 Initial catalog: TenantDataDB Username: rootuser Password: •••••••••••••••• Connection String Data Source=sqlrelay-vm;Initial Catalog=TenantDataDB;Integrated Security=False;User ID=demoUser;Password=@BuildHands0nLab2018;Application Name=UserTransactionsApp	Use the parameters from your "Workshop Sheet – Parameters".

Select the ‘**App Data**’ tab
Click ‘**Run**’

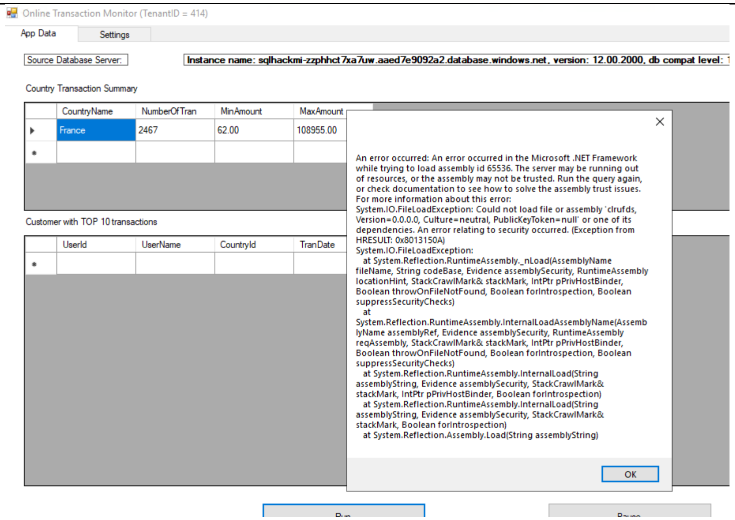
GOTCHA

If you get a long-winded error when you run the application it’s because the CLR assemblies don’t have the correct trust settings.

Run the 3 ALTER DATABASE statements below and try starting the application again.

After a few seconds you should see transactions start to appear.

Notice that the “Source Database Server” displayed at the top of the application shows the SQL Managed Instance FQDN.



Source Database Server: Instance name: sqlhackmi-zzphhct7xa7uw.aaed7e9092a2.database.windows.net, version: 12.00.2000, db compat level: 1

Country Transaction Summary

	CountryName	NumberOfTran	MinAmount	MaxAmount
▶	France	2467	62.00	108955.00
*				

Customer with TOP 10 transactions

	UserId	UserName	CountryId	TranDate
*				

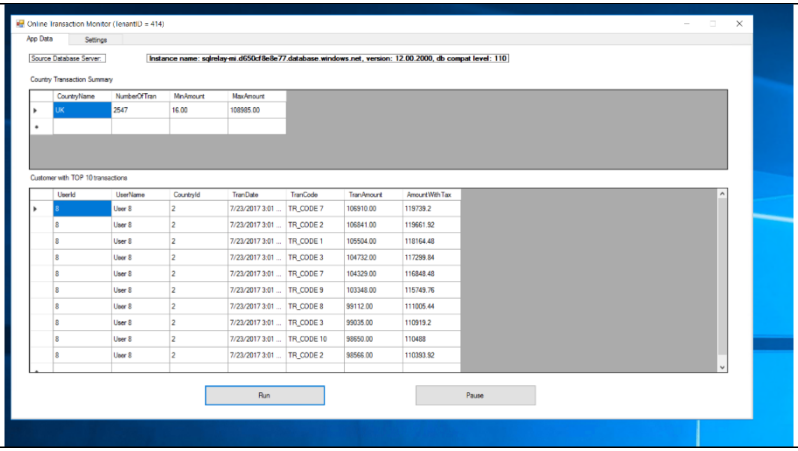
```
An error occurred: An error occurred in the Microsoft .NET Framework
while trying to load assembly id 65536. The server may be running out
of resources, or the assembly may not be trusted. Run the query again,
or check documentation to see how to solve the assembly trust issues.
For more information about this error:
System.IO.FileLoadException: Could not load file or assembly 'clrufds,
Version=0.0.0.0, Culture=neutral, PublicKeyToken=null' or one of its
dependencies. An error relating to security occurred. (Exception from
HRESULT: 0x8013150A)
System.IO.FileLoadException:
   at System.Reflection.RuntimeAssembly._nLoad(AssemblyName
fileName, String codeBase, Evidence assemblySecurity, RuntimeAssembly
locationHint, StackCrawlMark& stackMark, IntPtr pPrivHostBinder,
Boolean throwOnFileNotFound, Boolean forIntrospection, Boolean
suppressSecurityChecks)
   at
System.Reflection.RuntimeAssembly.InternalLoadAssemblyName(Assemb
lyName assemblyRef, Evidence assemblySecurity, RuntimeAssembly
reqAssembly, StackCrawlMark& stackMark, IntPtr pPrivHostBinder,
Boolean throwOnFileNotFound, Boolean forIntrospection, Boolean
suppressSecurityChecks)
   at System.Reflection.RuntimeAssembly.InternalLoad(String
assemblyString, Evidence assemblySecurity, StackCrawlMark&
stackMark, IntPtr pPrivHostBinder, Boolean forIntrospection)
   at System.Reflection.RuntimeAssembly.InternalLoad(String
assemblyString, Evidence assemblySecurity, StackCrawlMark&
stackMark, Boolean forIntrospection)
   at System.Reflection.Assembly.Load(String assemblyString)
```

```
-------------------------------------------------------
-- CHANGE BELOW TO YOUR TEAM NUMBER (REPLACE XX)
USE [TEAMXX_TenantDataDb]
GO
-------------------------------------------------------
EXEC dbo.sp_changedbowner 'sa'

alter database [TEAMXX_LocalMasterDataDB] set trustworthy on
go
alter database [TEAMXX_SharedMasterDataDB] set trustworthy on
go
alter database [TEAMXX_TenantDataDb] set trustworthy on
go
```

The application will generate simulated transactional data.

Notice how the ‘Source Database Server’ connection reflects the SQL Managed Instance proving that the database migration has been completed successfully.

Source Database Server:

Instance name: sqlrelay-mi.d650cf8e8e77.database.windows.net, version: 12.00.2000, db compat level: 110

Country Transaction Summary

CountryName	NumberOfTran	MinAmount	MaxAmount
UK	2547	16.00	108985.00

Customer with TOP 10 transactions

UserId	UserName	CountryId	TranDate	TranCode	TranAmount	AmountWithTax
8	User 8	2	7/23/2017 3:01 ...	TR_CODE 7	106910.00	119739.2
8	User 8	2	7/23/2017 3:01 ...	TR_CODE 2	106841.00	119661.92
8	User 8	2	7/23/2017 3:01 ...	TR_CODE 1	105504.00	118164.48
8	User 8	2	7/23/2017 3:01 ...	TR_CODE 3	104732.00	117299.84
8	User 8	2	7/23/2017 3:01 ...	TR_CODE 7	104329.00	116848.48
8	User 8	2	7/23/2017 3:01 ...	TR_CODE 9	103348.00	115749.76
8	User 8	2	7/23/2017 3:01 ...	TR_CODE 8	99112.00	111005.44
8	User 8	2	7/23/2017 3:01 ...	TR_CODE 3	99035.00	110919.2
8	User 8	2	7/23/2017 3:01 ...	TR_CODE 10	98650.00	110488
8	User 8	2	7/23/2017 3:01 ...	TR_CODE 2	98566.00	110393.92