

SQL Moderation Hack Database Migration Lab Step-by-step

V2.2

Contents

Migration architecture and Azure components 2

Generic Migration Content 3

1. Investigate the ‘Online Transaction Monitor’ legacy application 4

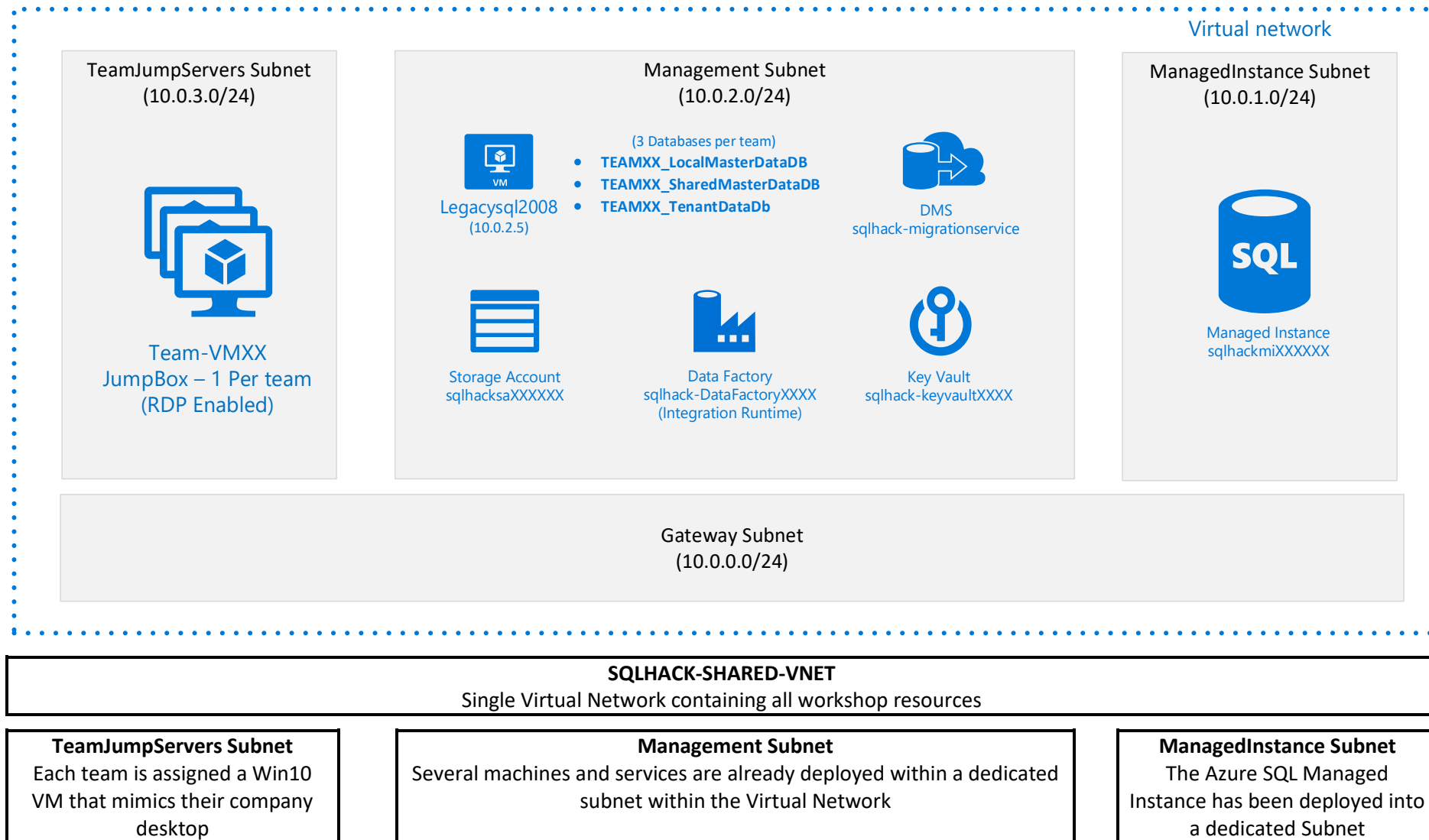
2. Assess the application databases for Azure SQL Database suitability using the Database Migration Assistant (DMA) 7

3. Use Azure Database Migration Service (DMS) to migrate the 3 application databases..... 17

4. Confirm application databases have been migrated to Azure SQL Managed Instance..... 29

5. Connect ‘Online Transaction Monitor’ App to Azure SQL DB Managed Instance 30

Migration architecture and Azure components

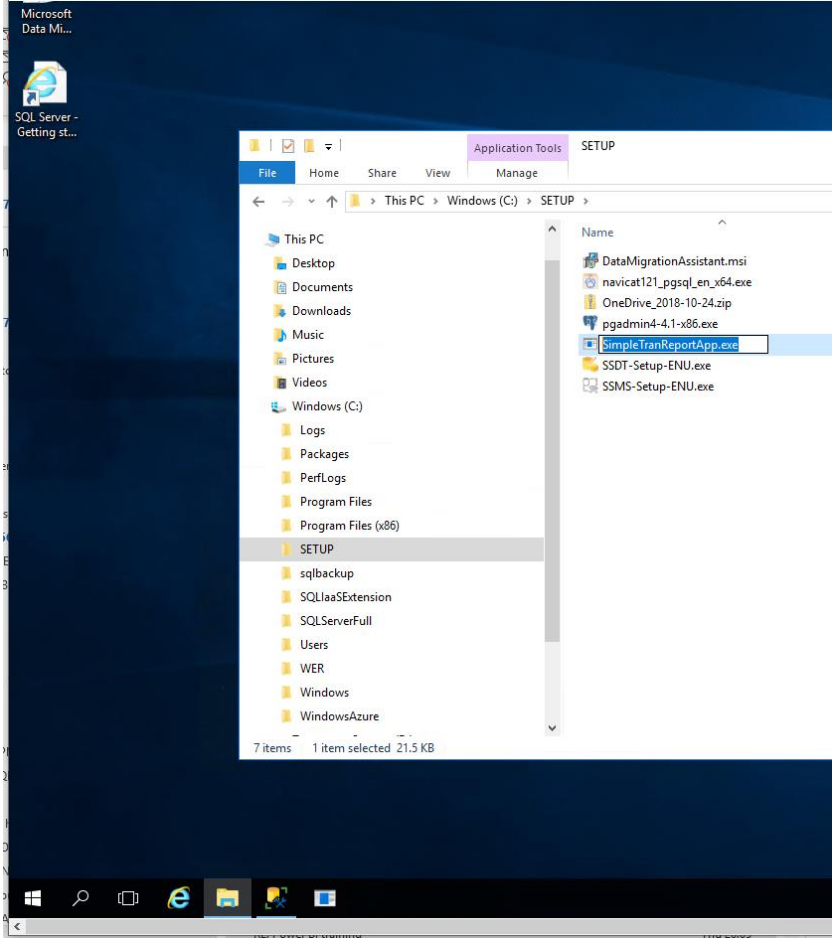


Generic Migration Content

Narrative	Notes
<i>Notes for outside of the workshop:</i> <i>Familiarise yourself with Microsoft migration tools and the Azure Database Migration Guide</i>	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. Investigate the 'Online Transaction Monitor' legacy 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
<p>We will set the sample application running to demonstrate how Azure Database Migration Services can be used to perform a migration of a database.</p> <p>RDP onto the Win10 management VM using the details from the "DB Migration Lab and Parameters.pdf"</p> <p>Run the 'SimpleTranReportApp' app which can be found in: C:_SQLHACK_\LABS\01-Data_Migration</p> <p>Or</p> <p>In Windows Explorer search for 'SimpleTranReportApp.exe'</p>		<p>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.</p>

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
[masked]

Change Connection String

Connection String

Data Source=sqlrelay-vm;Initial Catalog=TenantDataDB;Integrated Security=False;User ID=demoUser;Password=@BuildHandsOnLab2018;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.

SQL Modernisation Open Hack

Select 'App Data' tab and click the "Run" button.

After a few seconds transaction will start to appear in the application.

Online Transaction Monitor (TenantID = 414)

App Data Settings

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

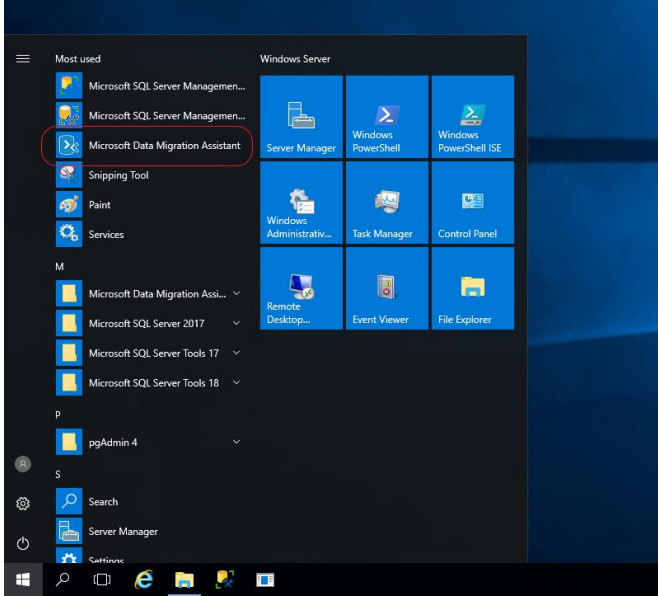
Userid	UserName	CountryId	TranDate	TranCode	TranAmount	AmountWithTax
19	User 19	3	7/23/2017 3:01 ...	TR_CODE 7	106899.00	119726.88
19	User 19	3	7/23/2017 3:01 ...	TR_CODE 5	105039.00	117643.68
19	User 19	3	7/23/2017 3:01 ...	TR_CODE 7	103876.00	116341.12
19	User 19	3	7/23/2017 3:01 ...	TR_CODE 3	102969.00	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

Run Pause

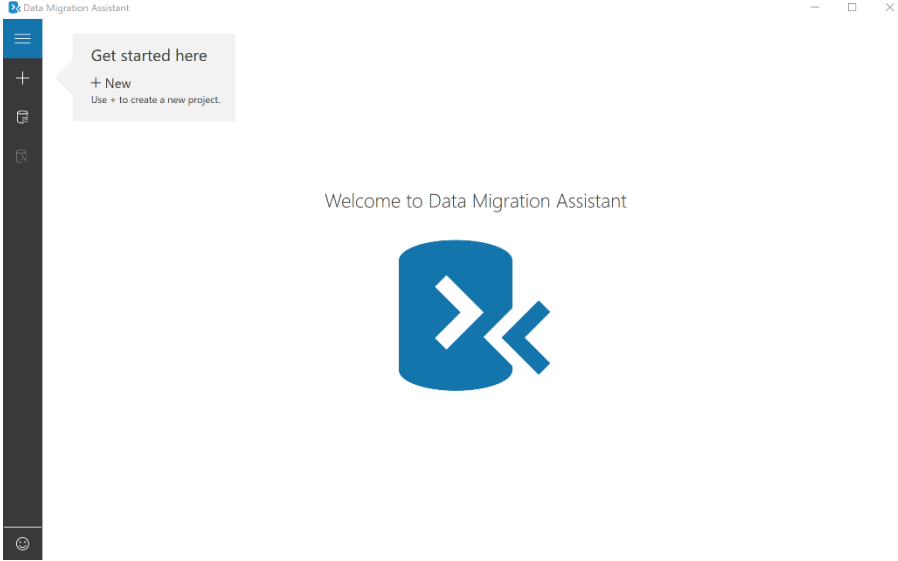
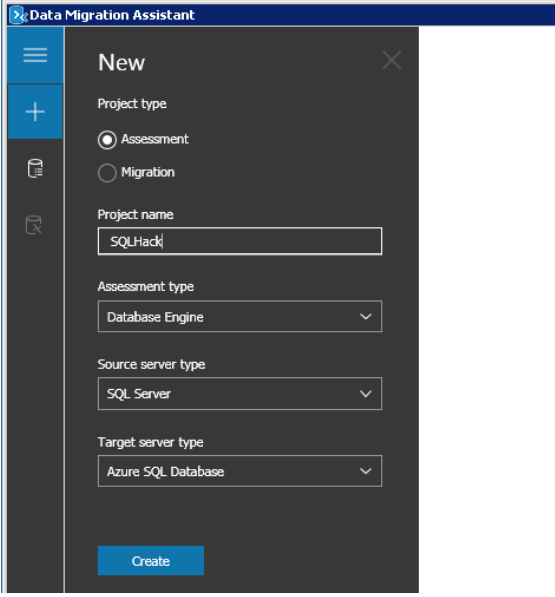
The application will generate simulated transactional data. Notice how the 'Source Database Server' information at the top of the app reflects the parameters given in the previous step.

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

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

Narrative	Screenshot	Notes
<p>We need to determine the suitability of the database(s) for migration to Azure. This includes checking for compatibility and feature support with Azure Database.</p> <p>You should already have an RDP session open to your teams Win10 Management VM, if so run DMA from the Start menus or Desktop icon.</p>	 <p>The screenshot shows the Windows Server Start menu. The 'Most used' section on the left lists several applications, with 'Microsoft Data Migration Assistant' circled in red. The right pane shows various system tools like Server Manager, Windows PowerShell, and Task Manager.</p>	<p>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.</p>

SQL Modernisation Open Hack

<p>You should see this screenshot to the right.</p> <p>Select the “+” to create a new assessment project</p>		
<p>Select/Enter the following details:</p> <p>Project name: Workshop1</p> <p>Assessment type: Database Engine</p> <p>Source server type: SQL Server</p> <p>Target server type: Azure SQL Database</p> <p>Click ‘Create’</p>		<p>Our first project assessment assumes we will be migrating to Azure SQL DB, so the options shown in the screenshot need to be selected.</p>

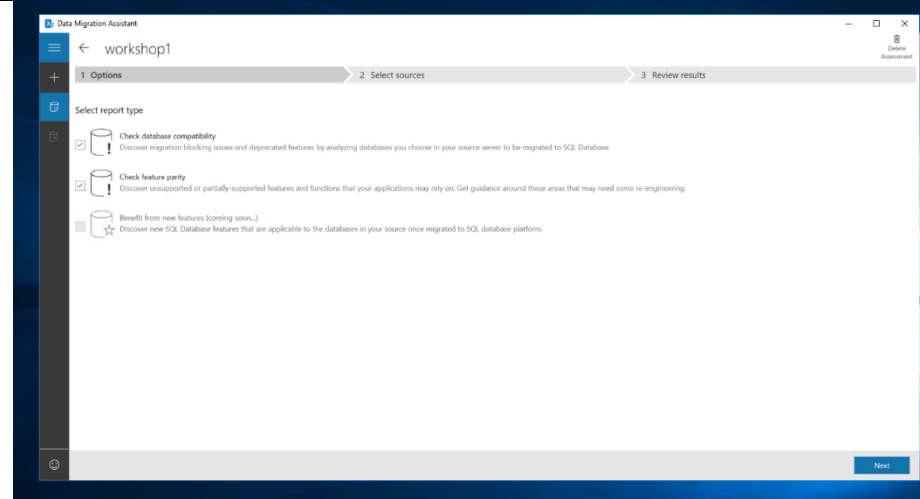
SQL Modernisation Open Hack

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:

LEGACYSQL2008

Authentication Type:

SQL Server Authentication

Username:

Demouser

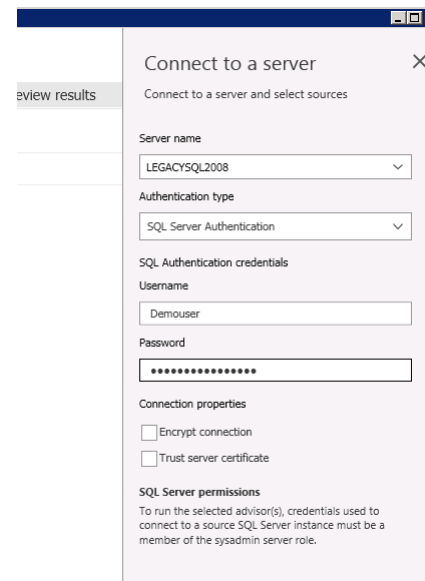
Password:

Demo@pass1234567

Untick "Encrypt connection"

Click '**Connect**'

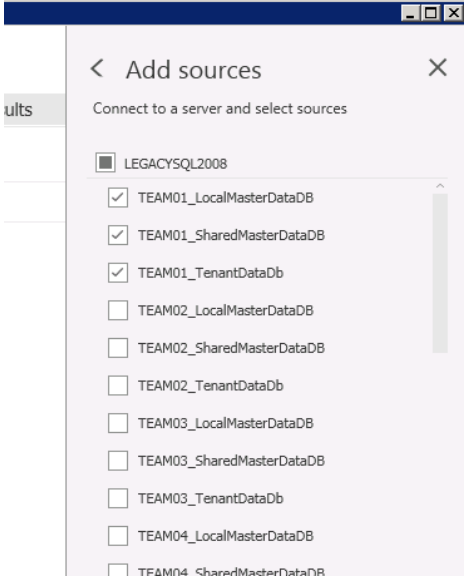
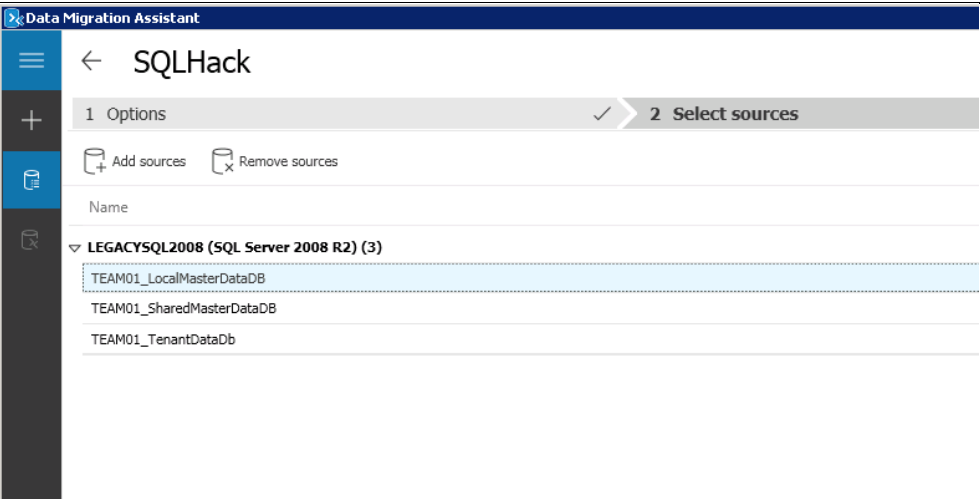
**If you get an error logging in
check that the Win10 keyboard
is set to UK English not US**



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.

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

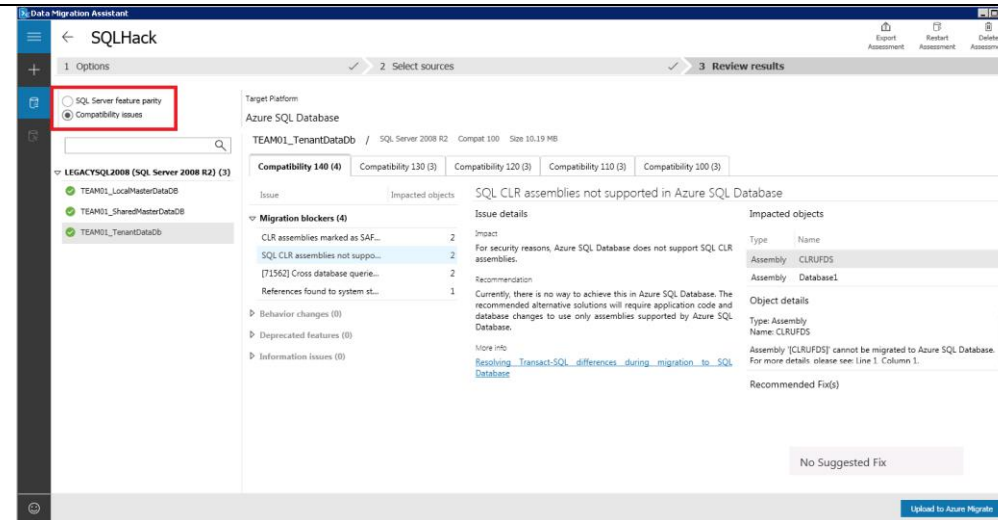
DMA will now show the results of the assessment using 2 separate reports:

‘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 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).

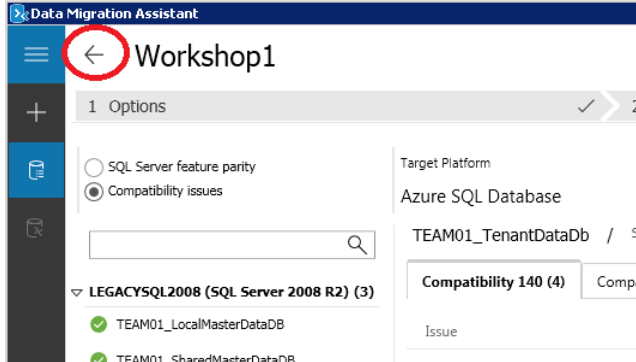
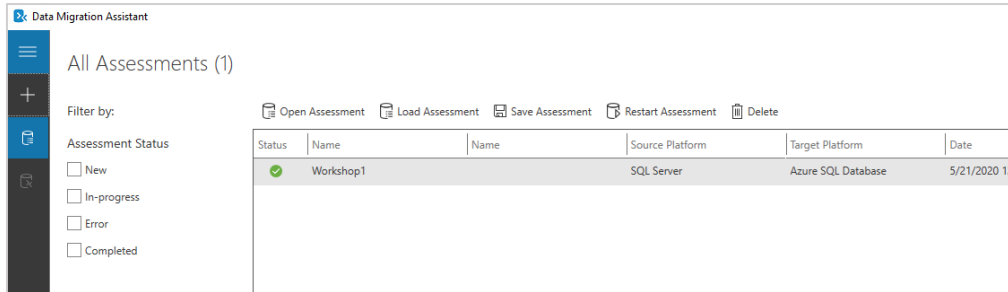


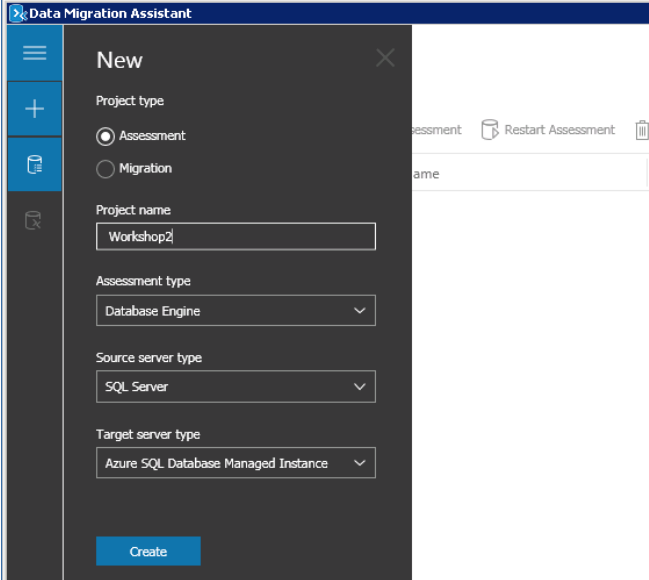
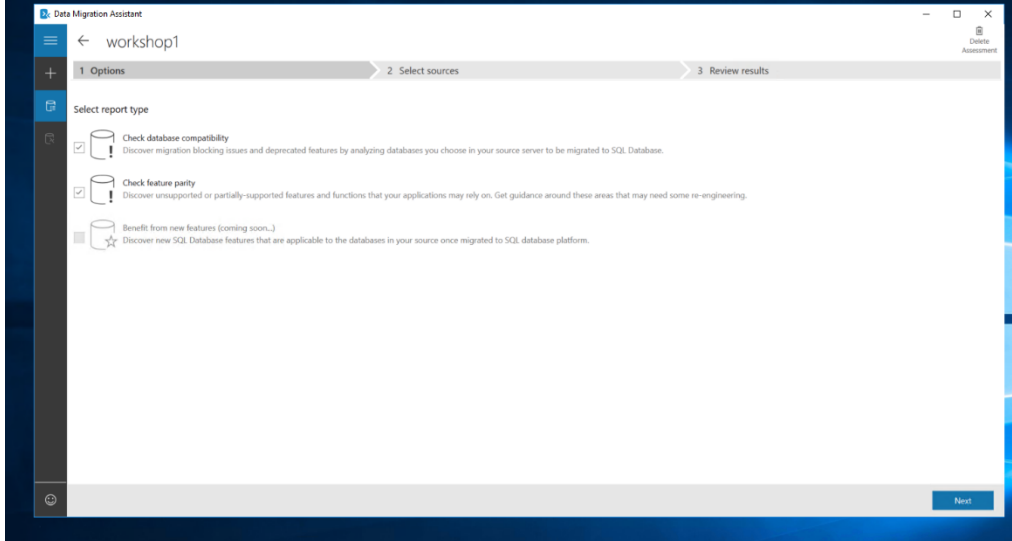
Note: Toggle the parity and compatibility issues radio button (top left) to switch between the 2 reports.

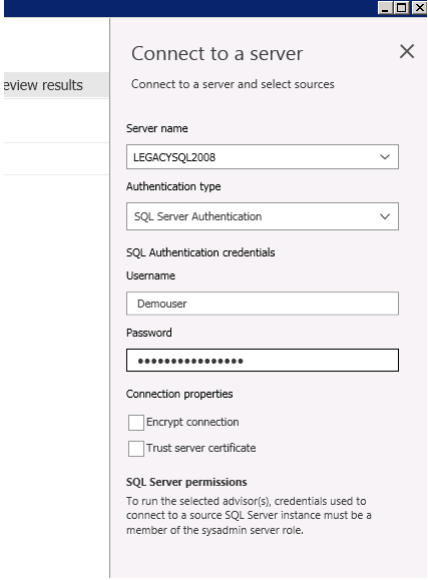
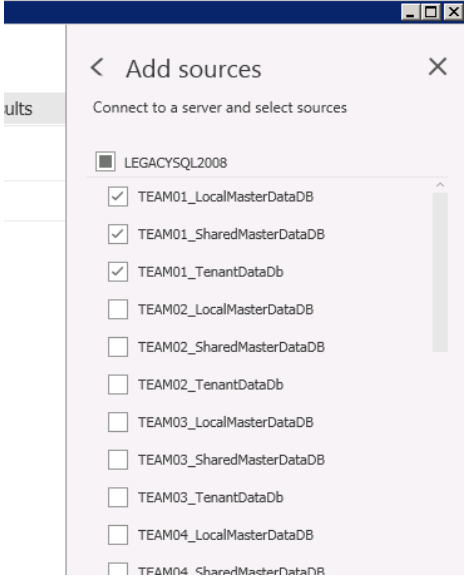
‘SQL Server feature parity’ shows what features are not supported in the target data source. Under the ‘Details’ and ‘Databases’ sections on the right you will find remedial action that are required and the databases impacted.

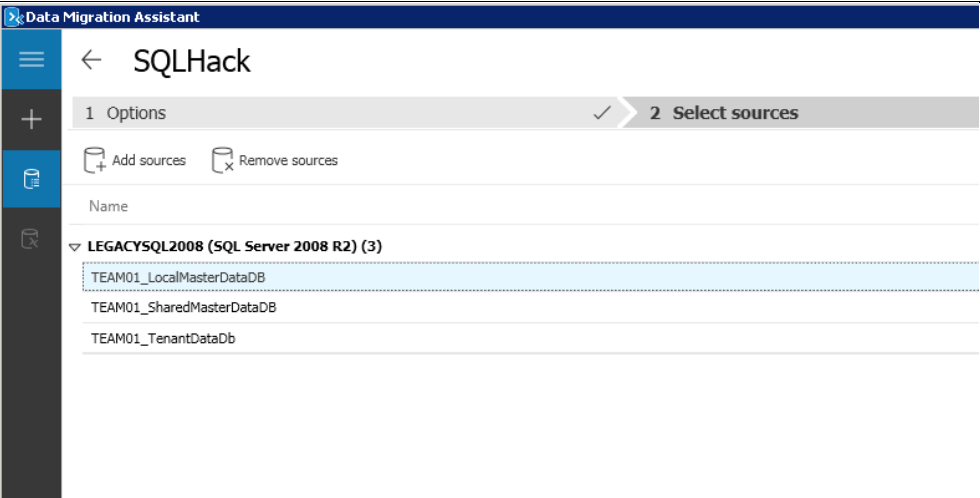
‘Compatibility Issues’ shows, over the compatibility tabs, issues that need to be addressed to permit the database(s) to run, in the chosen compatibility level (e.g. 140, 130, 120, 110).

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

<p>Once you’re reviewed the assessment click the back arrow to see a list of current DMA projects.</p>														
	<p>Because we need to migrate CLR Stored Procs, we need to repeat the assessment with Azure SQL DB Managed Instance as the target to see if it’s compatible.</p>													
<p>You should see this screenshot to the right.</p> <p>Select the “+” to create a new assessment project.</p>	 <table><tr><th>Status</th><th>Name</th><th>Name</th><th>Source Platform</th><th>Target Platform</th><th>Date</th></tr><tr><td>✓</td><td>Workshop1</td><td></td><td>SQL Server</td><td>Azure SQL Database</td><td>5/21/2020 12</td></tr></table>	Status	Name	Name	Source Platform	Target Platform	Date	✓	Workshop1		SQL Server	Azure SQL Database	5/21/2020 12	
Status	Name	Name	Source Platform	Target Platform	Date									
✓	Workshop1		SQL Server	Azure SQL Database	5/21/2020 12									

<p>Select/Enter the following details:</p> <p>Project name: Workshop2</p> <p>Assessment type: Database Engine</p> <p>Source server type: SQL Server</p> <p>Target server type: Azure SQL Database Managed Instance</p> <p>Click 'Create'</p>		<p>Our 2nd assessment project assumes we will be migrating to Azure SQL DB Managed Instance, so the options shown in the screenshot need to be selected.</p>
<p>Select the assessment checks (Report Type) to be made:</p> <p>Check database compatibility</p> <p>Check feature parity</p> <p>Click 'Next'</p>		<p>As we saw previously DMA can test for both database compatibility and feature parity compliance against the chosen target.</p> <p>As before we will assess all the databases against all of the tests.</p>

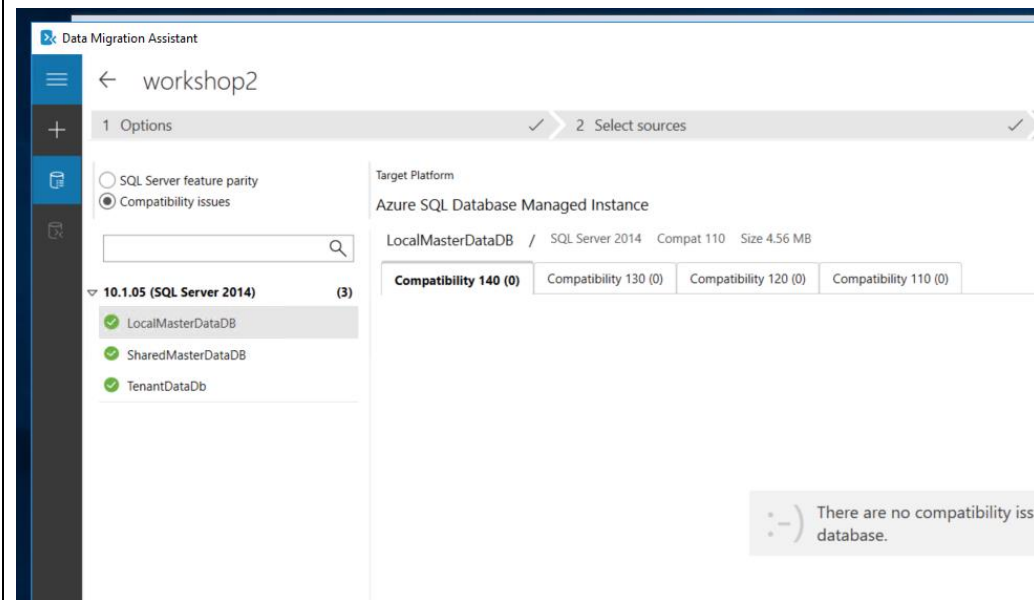
<p>Enter the source/legacy SQL details:</p> <p>Server Name: LEGACYSQL2008</p> <p>Authentication Type: SQL Server Authentication</p> <p>Username: Demouser</p> <p>Password: Demo@pass1234567</p> <p>Untick “Encrypt connection”</p> <p>Click ‘Connect’</p>		<p>When performing this within your own subscription you will enter the host, authentication and connection types according to your company guidelines and practices.</p> <p><i>Bear in mind that DMA needs to connect to a source SQL Server using an account that belongs to the sysadmin role.</i></p> <p>As this document is produced within a workshop environment Active Directory, Certificates and encryption has not been setup.</p>
<p>Select only the 3 database used by your ‘Online Transaction Monitor’ app. These will have a TEAMXX prefix where XX should be replaced by your team number.</p> <p>TEAMXX_LocalMasterDataDb</p> <p>TEAMXX_SharedMasterDb</p> <p>TEAMXX_TenantDataDb</p> <p>Click ‘Add’ to add them to the assessment.</p>		<p>DMA will show all databases located on the Source host and display them so you can decide which ones to include in this assessment project.</p> <p>Note that you can assess multiple databases at the same time.</p>

<p>You should now see the screen on the right with the relevant TEAMXX databases listed.</p> <p>Select 'Start Assessment'</p>		<p>Note: DMA allows you to either 'Add' or 'Remove' additional data sources as needed at this point.</p> <p>Also note that DMA has identified what compatibility level each source database is running under.</p>

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.



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

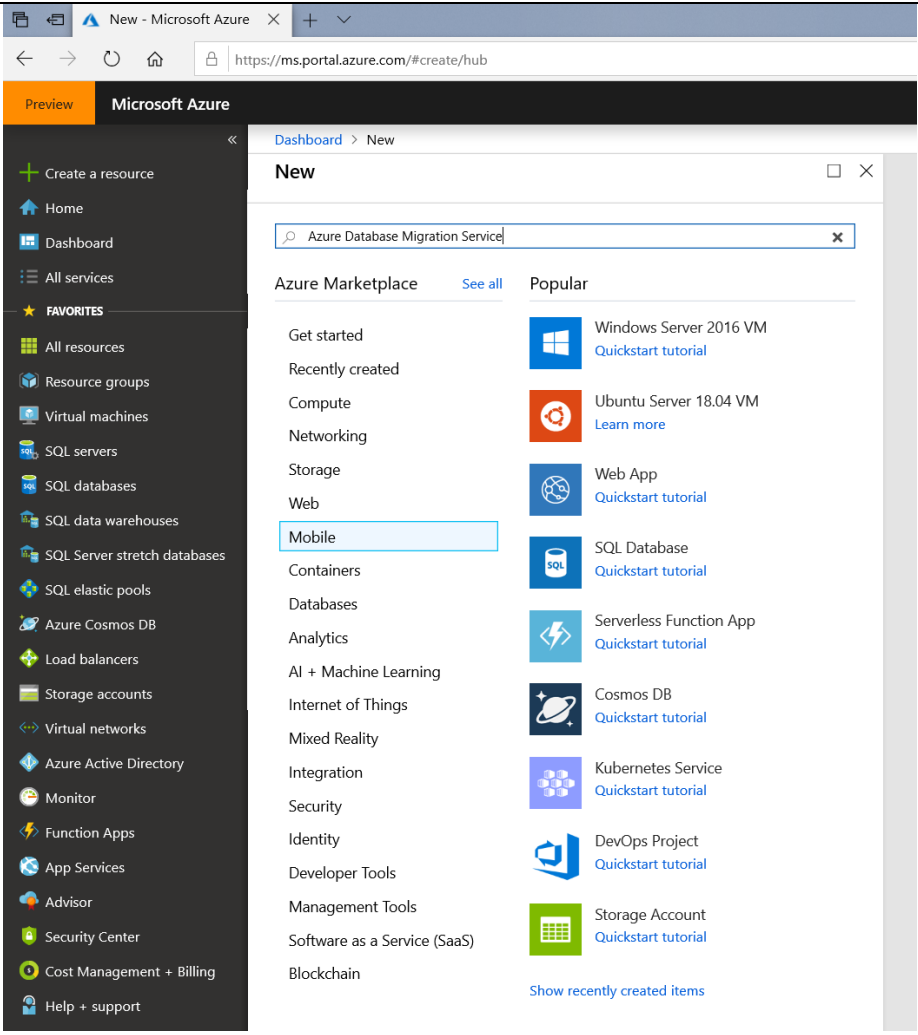
'SQL Server feature parity' shows what features are not supported in the target datasource. Under 'Details' and 'Databases' you will find remedial action that are required and the databases impacted.

'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
<p>We will be using Azure Database Migration Service (DMS) to migrate the legacy SQL2008 databases to Azure.</p> <p>For the workshop DMS will already been provisioned as it can take 20-30mins to be deployed.</p> <p>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.</p>		<p>DMS is provisioned as a service which hosts & runs multiple migration Projects. Each Project is responsible for migrating one or more databases.</p> <p>Although a Project can migrate multiple databases each Project can only migrate databases from a single source host to a single target destination.</p> <p>In this lab we will use a single Project to migrate 3 databases from the same legacy SQL2008 host to Azure SQL Managed Instance.</p> <p>DMS can host and run different types of database migration Projects under the same instance e.g. separate Project for separate source servers.</p> <p>Overview of DMS: Azure Database Migration Service</p> <p>DMS tutorials: https://docs.microsoft.com/en-us/azure/dms/</p>

For this workshop:

On your Win10 VM open Edge browser and got to:

[HTTPS://portal.azure.com](https://portal.azure.com)

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

Username:

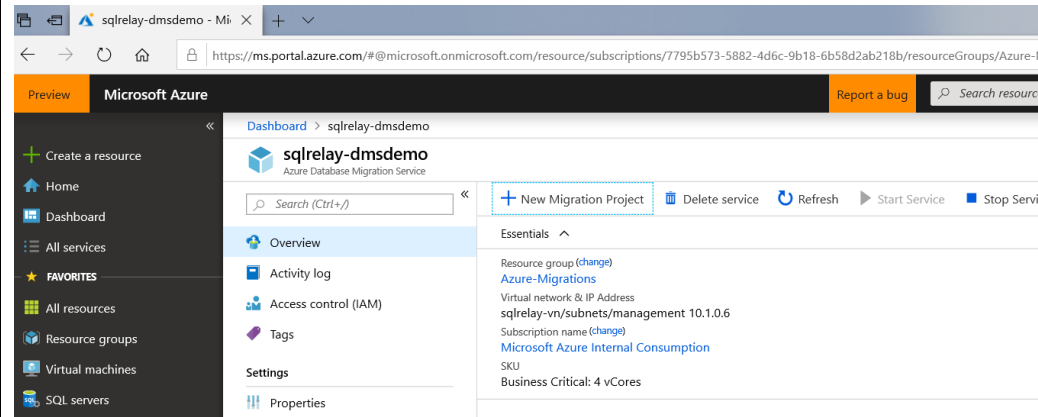
SQLHACK_TEAMXXXxx@OTAPRD672ops.onmicrosoft.com

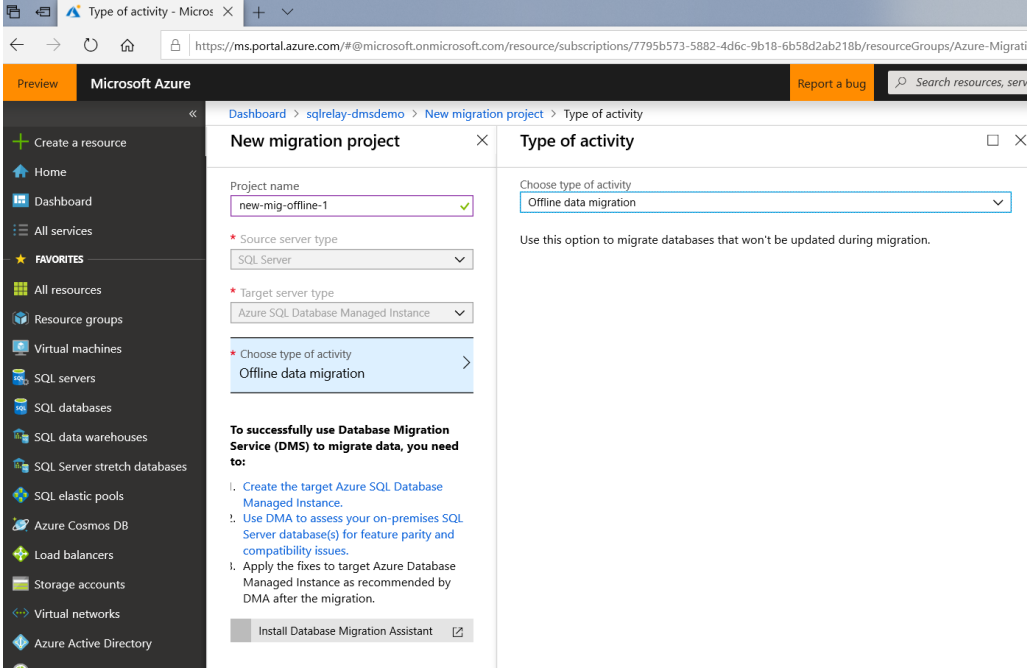
Password:

<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'**



<p>On the “New migration project” blade use the following settings:</p> <p>Project name: TEAMXX_migration <i>(replace XX with your team number)</i></p> <p>Source server type: SQL Server</p> <p>Target Server type: Azure SQL Database Managed Instance</p> <p>Choose type of activity: Offline data migration</p> <p>Click ‘Save’</p> <p>Click ‘Create and run activity’</p>	 <p>The screenshot shows the 'New migration project' blade in the Azure portal. The 'Project name' field contains 'new-mig-offline-1'. The 'Source server type' is set to 'SQL Server'. The 'Target server type' is set to 'Azure SQL Database Managed Instance'. The 'Choose type of activity' dropdown is set to 'Offline data migration'. A sidebar on the left shows the navigation menu with 'SQL servers' selected. A bottom section lists steps to successfully use Database Migration Service (DMS) to migrate data:</p> <ol style="list-style-type: none"> 1. Create the target Azure SQL Database Managed Instance. 1. Use DMA to assess your on-premises SQL Server database(s) for feature parity and compatibility issues. 1. Apply the fixes to target Azure Database Managed Instance as recommended by DMA after the migration. <p>There is a button labeled 'Install Database Migration Assistant' at the bottom.</p>	<p>DMS can perform two types of database migrations:</p> <ul style="list-style-type: none"> - Offline - Online <p>Offline migrations use backup files. The backups can be provided to DMS or DMS can create the backup as part of a project.</p> <p>Whilst the simplest to perform, taking the backup, moving it to Azure and restoring it can cause significant downtime.</p> <p>Online migrations use a replication or log shipping approach to keep the source & target in sync. Whilst more complex it significantly reduces database downtime.</p>

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

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**’.

DMS will perform a connection test.

The screenshot displays the Microsoft Azure Migration Wizard interface. The left sidebar shows the navigation menu with options like 'Create a resource', 'Home', 'Dashboard', 'All services', and 'FAVORITES'. The main area shows the 'Migration Wizard' steps. Step 1, 'Select source', is the active step and is marked with a green checkmark. The 'Migration source detail' pane on the right contains the following information:

- Source SQL Server instance name:** 10.0.2.5
- Authentication type:** SQL Authentication (dropdown menu)
- User Name:** Demouser
- Password:** (masked with dots)
- 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.

The screenshot displays the Microsoft Azure Migration Wizard interface. The breadcrumb navigation shows: Dashboard > sqlhack-migrationservice > Migration Wizard > Migration target details. The left sidebar contains navigation options: Create a resource, Home, Dashboard, All services, FAVORITES, Cost Management + Billing, All resources, Resource groups, SQL managed instances, SQL databases, Analysis Services, Virtual machines, Azure Active Directory, Azure Cosmos DB, Storage accounts, Virtual networks, and Monitor. The main content area shows a list of steps: 1 Select source (checked), 2 Select target (checked and highlighted), 3 Select databases, 4 Select logins, 5 Configure migration settings, and 6 Summary. The 'Migration target details' panel on the right contains the following fields: Target server name (sqlhackmi-zzphct7xa7uw.aad7e9092a2.da...), Authentication type (SQL Authentication), User Name (Demouser), and Password (masked with dots).

STEP 3: Select Databases

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

TEAMXX_LocalMasterDataDb

TEAMXX_SharedMasterDb

TEAMXX_TenantDataDb

(replace XX with your team number)

Select 'Save'

Microsoft Azure Migration Wizard - 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'

Microsoft Azure Migration Wizard - Select logins

Source server name: 10.0.2.5

Search to filter items...

SOURCE LOGINS (27)

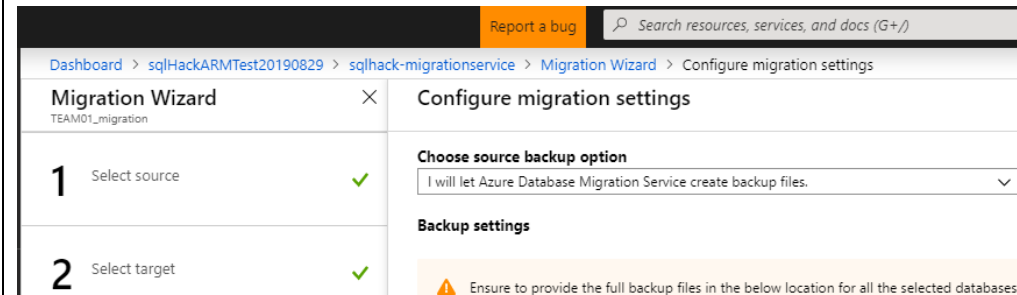
LOGIN TYPE	DEFAULT DATABASE	STATUS	
<input checked="" type="checkbox"/> TEAM09	SQL	master	Enabled
NT SERVICE\MSSQLSERVER	Windows group	master	Enabled
TEAM07	SQL	master	Enabled
NT SERVICE\SQLSERVERAGENT	Windows group	master	Enabled
TEAM20	SQL	master	Enabled
sa	SQL	master	Disabled
TEAM11	SQL	master	Enabled
TEAM18	SQL	master	Enabled
LEGACYSQL2008\DemoUser	Windows	master	Enabled
##MS_PolicyTsqlExecutionLogin##	SQL	master	Disabled
TFAM15	SQL	master	Enabled

Step 5a: Configure migration Settings

(Source Backup Option)

We are running an offline migration which will utilise backups of the DBs that are being migrated.

We want DMS to perform the backups, so select this option from the “Choose source backup option” (as shown).



DMS can use backup files provided or take the backups as part of the migration activity.

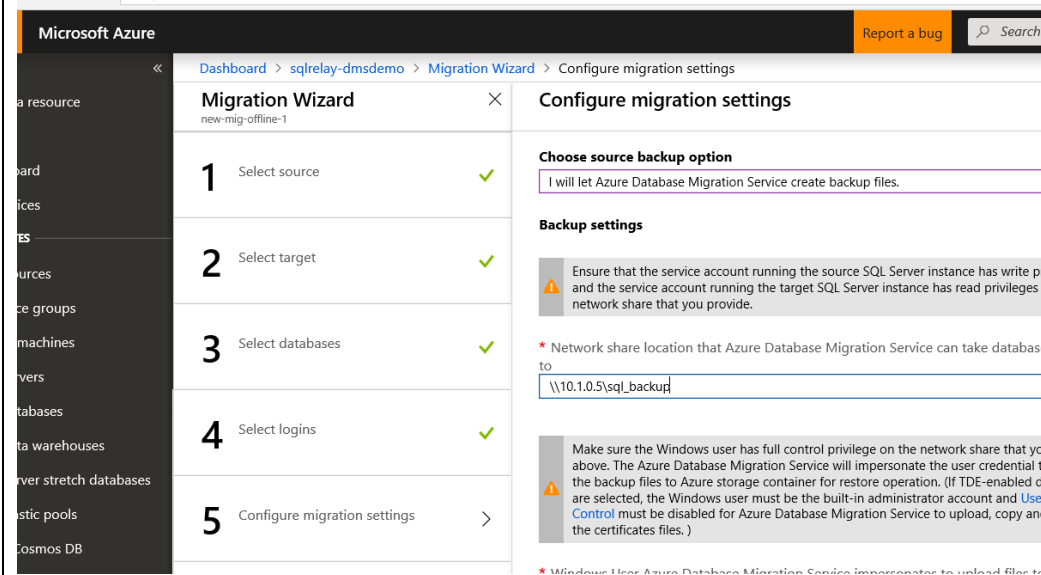
Step 5b: Configure migration Settings

(Backup location)

We can now enter the Windows share that the source server will write the database backups to.

Network Share:

\\LEGACYSQL2008\FILESHARE



Step 5c: Configure migration Settings

(Windows User for DMS)

Next 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

Microsoft Azure

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

Report a bug Search resources, serv

resource <<

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

Sqlrelay2014rootuser

Password

..... ✓

Step 5d: Configure migration Settings

(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 during the restore process.

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.

4 Select logins ✓

5 Configure migration settings ✓

6 Summary >

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://sqlhacksazphct7xa7uw.blob.core.windows.net/sqlhack?sp=racwdl&st=2019-08-3...

▼ Advanced settings

Save

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 and run migration)**

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)

Select ‘**Run migration**’

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 option

☐ Do not validate my database(s)

☒ Validate my database(s)

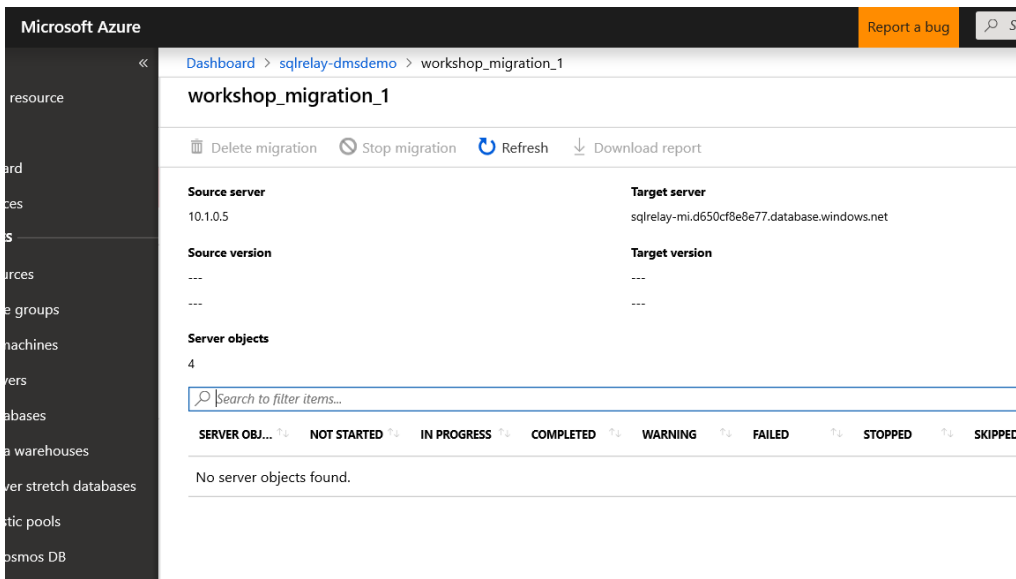
Validation options:
☐ Query correctness ⓘ

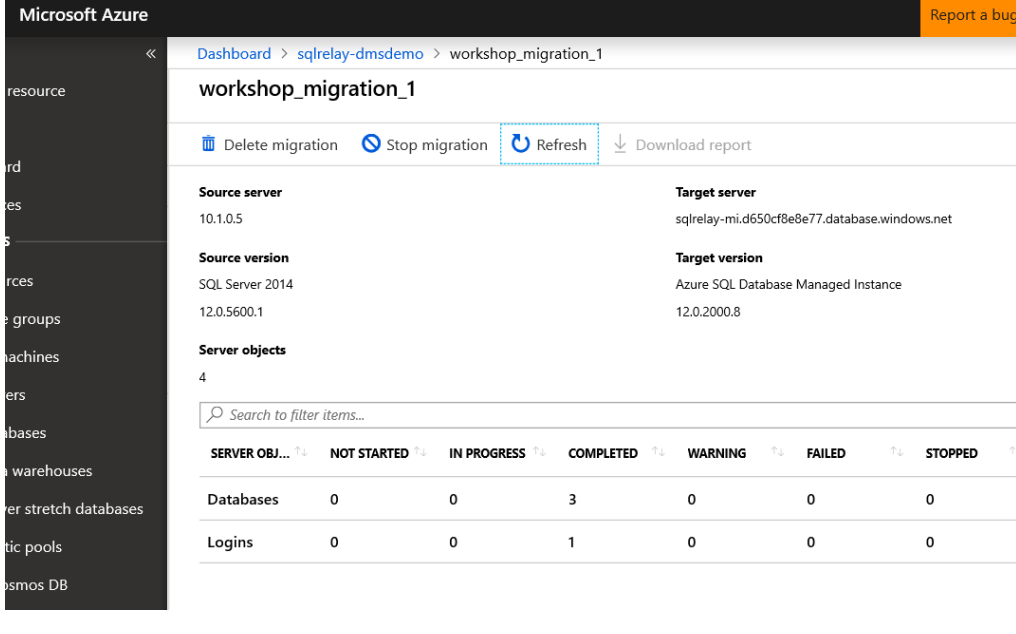
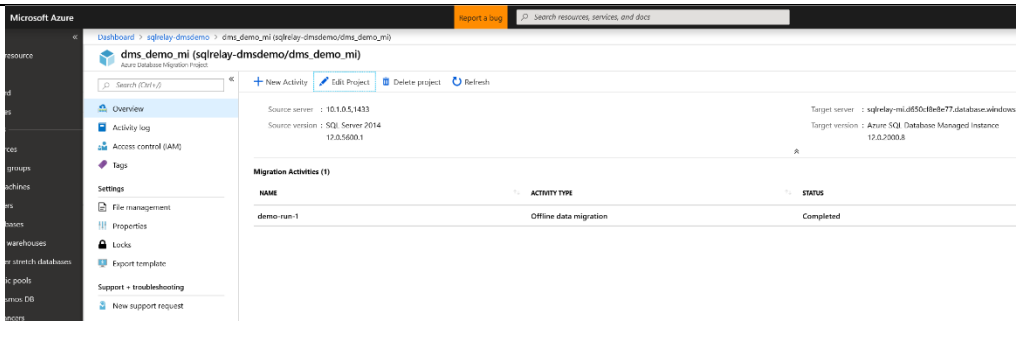
Validate my databases option

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.

<p>DMS will now run the migration activity.</p> <p>Initially this screen will be displayed.</p> <p>Select 'Refresh' to monitor the progress of your migration.</p> <p>Notice the database counts under the following columns as you keep pressing 'Refresh':</p> <p>“IN PROGRESS” “COMPLETED” “FAILED”</p>		<p>During the migration process you can monitor the creation of the 3 backup files by opening the fileshare \\LEGACYSQL2008\FILESHARE</p> <p>Also note that DMS clean-up after itself & deletes the backups from the fileshare once they have been copied up the Storage Account.</p>
--	--	---

<p>Under “COMPLETED”, when the number of databases says “3” the migration activity has completed.</p>	 <p>Microsoft Azure</p> <p>Dashboard > sqlrelay-dmsdemo > workshop_migration_1</p> <p>workshop_migration_1</p> <p>Delete migration Stop migration Refresh Download report</p> <p>Source server: 10.1.0.5 Source version: SQL Server 2014 12.0.5600.1 Target server: sqlrelay-mi.d650cf8e77.database.windows.net Target version: Azure SQL Database Managed Instance 12.0.2000.8 Server objects: 4</p> <p>Search to filter items...</p> <table><tr><th>SERVER OBJ...</th><th>NOT STARTED</th><th>IN PROGRESS</th><th>COMPLETED</th><th>WARNING</th><th>FAILED</th><th>STOPPED</th></tr><tr><td>Databases</td><td>0</td><td>0</td><td>3</td><td>0</td><td>0</td><td>0</td></tr><tr><td>Logins</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr></table>	SERVER OBJ...	NOT STARTED	IN PROGRESS	COMPLETED	WARNING	FAILED	STOPPED	Databases	0	0	3	0	0	0	Logins	0	0	1	0	0	0	<p>If there any warnings, errors or skipped databases they will have a database count under the corresponding headings in the status page.</p> <p>It is assumed, for the workshop, that all three databases have migrated successfully.</p>
SERVER OBJ...	NOT STARTED	IN PROGRESS	COMPLETED	WARNING	FAILED	STOPPED																	
Databases	0	0	3	0	0	0																	
Logins	0	0	1	0	0	0																	
<p>Close the migration activity.</p> <p>On the migration project blade notice that your migration activity is displayed.</p>	 <p>Microsoft Azure</p> <p>Dashboard > sqlrelay-dmsdemo > dms_demo_mi (sqlrelay-dmsdemo/dms_demo_mi)</p> <p>dms_demo_mi (sqlrelay-dmsdemo/dms_demo_mi)</p> <p>Source server: 10.1.0.5,1433 Source version: SQL Server 2014 12.0.5600.1 Target server: sqlrelay-mi.d650cf8e77.database.windows.net Target version: Azure SQL Database Managed Instance 12.0.2000.8</p> <p>Migration Activities (1)</p> <table><tr><th>NAME</th><th>ACTIVITY TYPE</th><th>STATUS</th></tr><tr><td>demo-run-1</td><td>Offline data migration</td><td>Completed</td></tr></table>	NAME	ACTIVITY TYPE	STATUS	demo-run-1	Offline data migration	Completed	<p>DMS keeps a history of activity runs for migration projects. A migration activity can be edited and ran again.</p>															
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:

Demouser

Password:

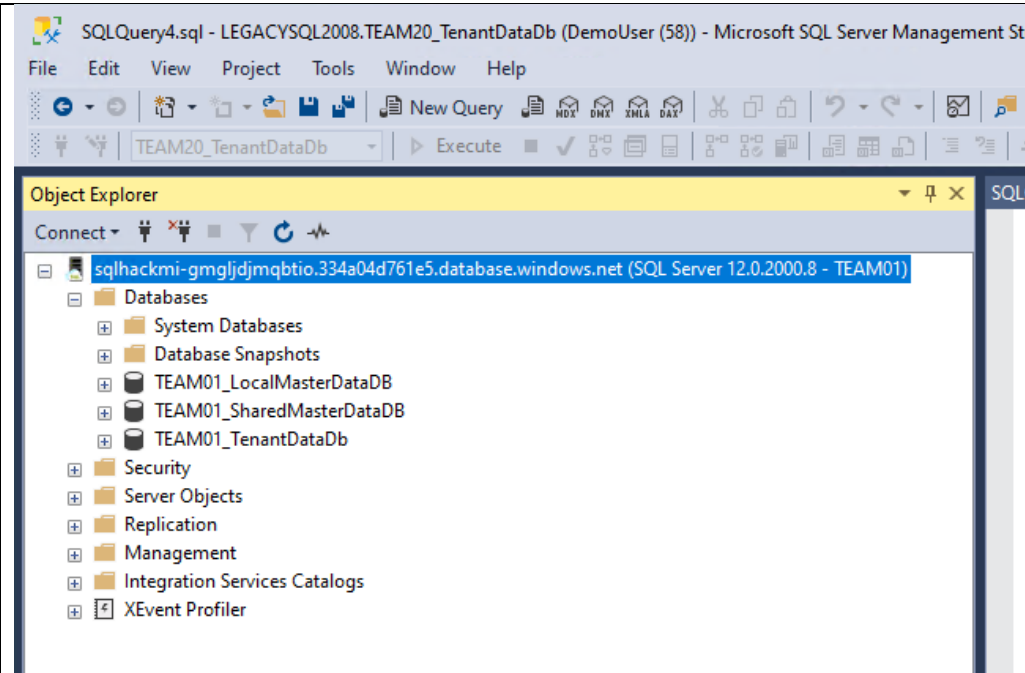
Demo@pass1234567

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

TEAMXX_LocalMasterDataDb

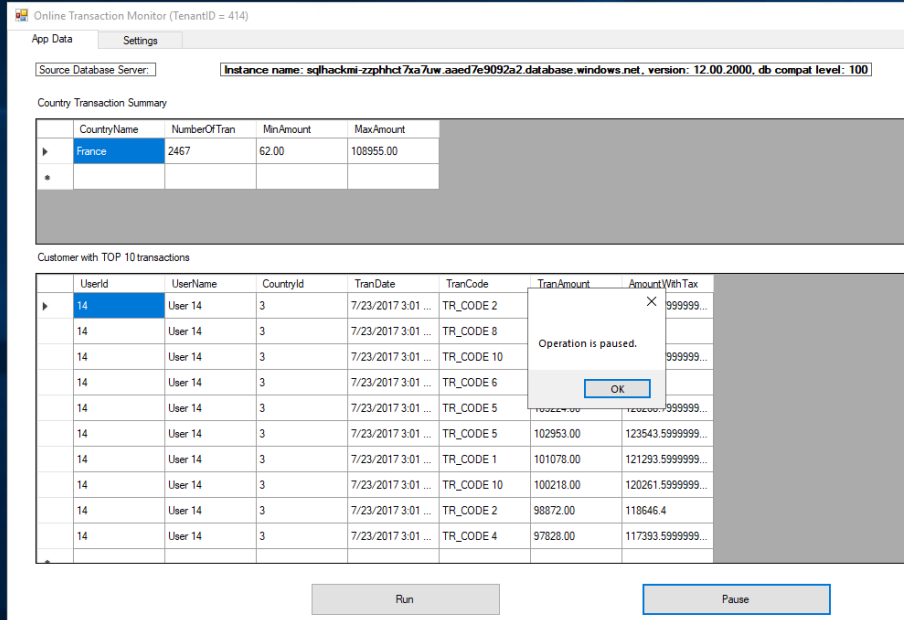
TEAMXX_SharedMasterDb

TEAMXX_TenantDataDb



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
<p>On your team Win10 management VM run the SimpleTranReportApp application.</p> <p>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'</p>		

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

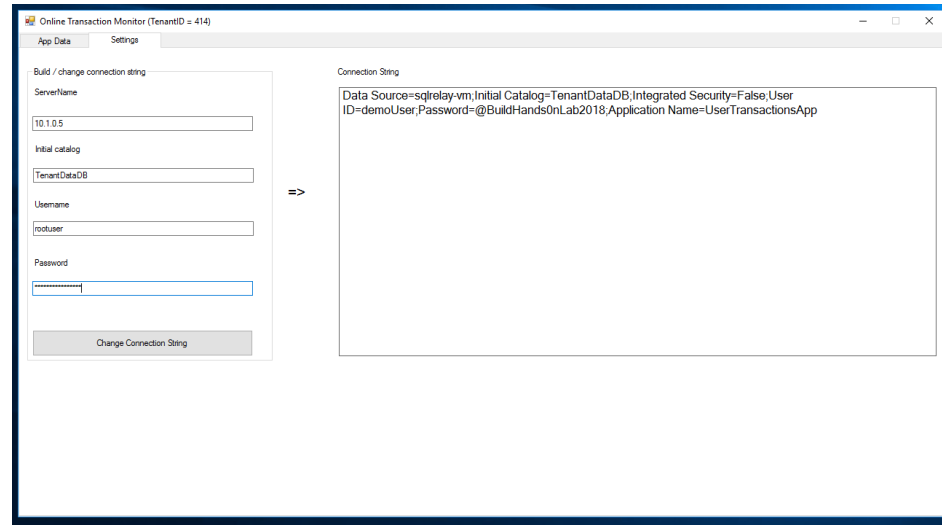
UserName:

TEAMXX

Password:

TEAMXX

Click '**Change Connection String**' to apply these new settings.



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.

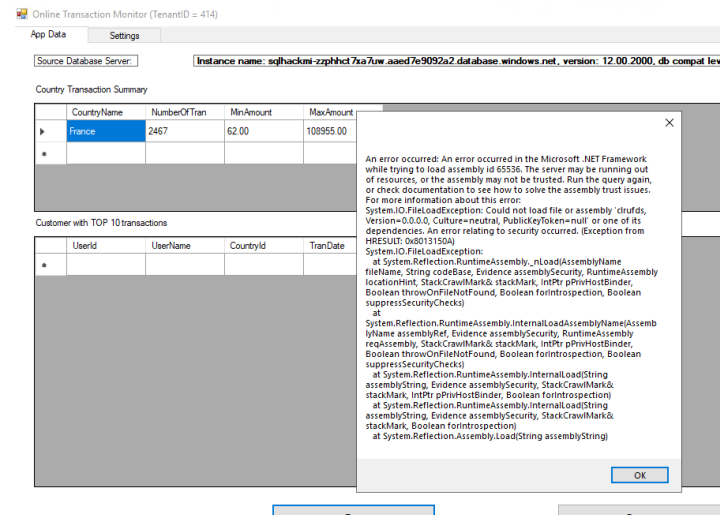
Run the 3 ALTER DATABASE statements below and try starting the application again.

You can find a skeleton script with these statements already prepared at:

**C:_SQLHACK_\LABS\
01-Data_Migration\
Migration Helper Script.txt**

Now go back to the application and try running it 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.



```
-- 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.

Online Transaction Monitor (ServerID = 414)

App Data Settings

Source Database Server: [Instance name: sqlrelay-ws-01502f6c177 database.windows.net, version: 12.00.2000, db compat level: 110]

Country Transaction Summary

CountryName	NumberOfTran	MinAmount	MaxAmount
UK	2547	15.00	105985.00

Customer with TOP 10 transactions

Userid	UserName	CountryId	TransDate	TransCode	TransAmount	AmountWithTax
1	User 0	2	7/23/2017 3:01	TR_CODE 7	106910.00	119739.2
8	User 0	2	7/23/2017 3:01	TR_CODE 2	100841.00	119661.92
8	User 0	2	7/23/2017 3:01	TR_CODE 1	105004.00	119164.48
8	User 0	2	7/23/2017 3:01	TR_CODE 3	104732.00	117299.84
8	User 0	2	7/23/2017 3:01	TR_CODE 7	104329.00	118648.48
8	User 0	2	7/23/2017 3:01	TR_CODE 9	103348.00	119749.76
8	User 0	2	7/23/2017 3:01	TR_CODE 8	99112.00	111005.44
8	User 0	2	7/23/2017 3:01	TR_CODE 3	99038.00	110919.2
8	User 0	2	7/23/2017 3:01	TR_CODE 10	88930.00	110488
8	User 0	2	7/23/2017 3:01	TR_CODE 2	88566.00	110393.92

Run Pause