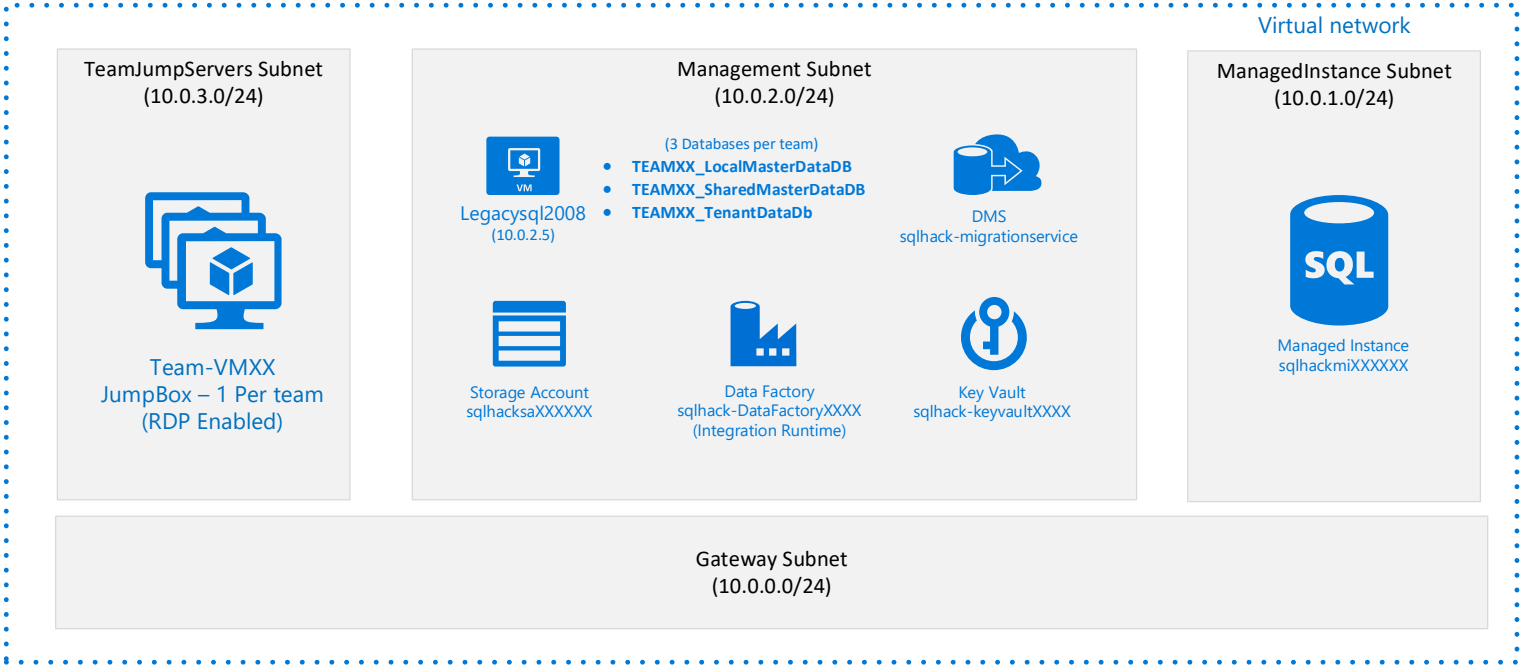
SQL Moderation Hack Database Migration Lab Step-by-step

Formatted: Font: (Default) +Body (Calibri), 11 pt, Font color: Auto

Contents

Migration architecture and Azure components

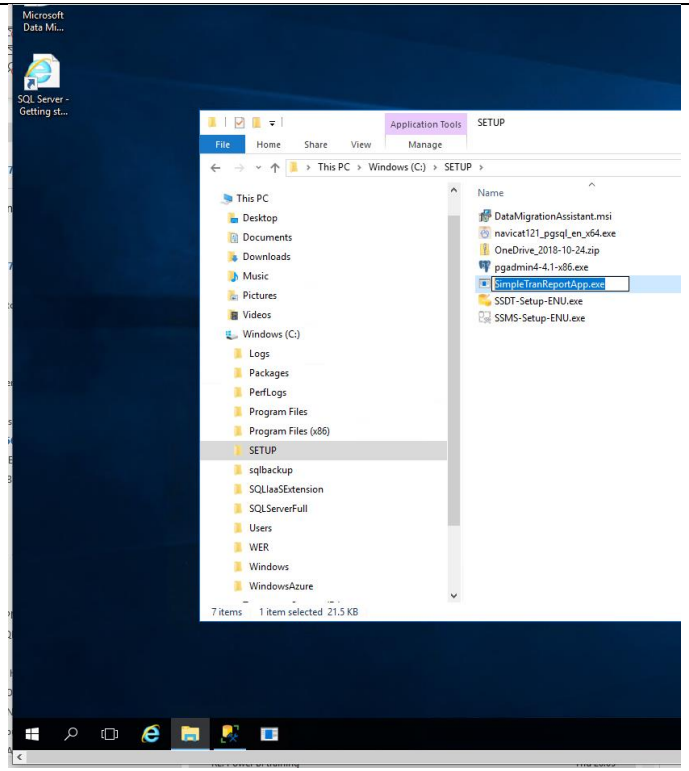


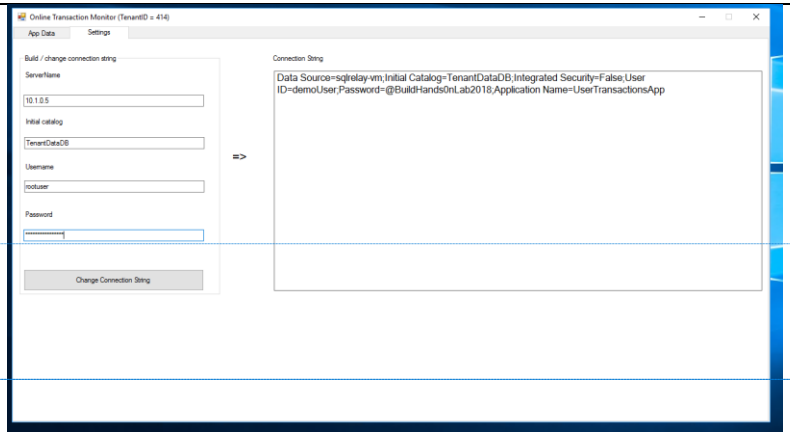
Generic Migration Content

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 an "on-line" migration of a running database with minimal downtime. ~~the simplicity "friction free" way to migrate an existing application to a Cloud environment.~~ ~~Connect~~ RDP onto the Win10 management ~~Windows management~~ VM using the **IP Address from Labs and Parameters" doc** On the Task Bar– select 'SimpleTranReportApp' Or Windows Explorer - 'SimpleTranReportApp.exe'		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 ~~Enter~~ the following parameters into the fields identified: **ServerName:** **LEGACYSQL2008** ~~**SourceSQLhostX**~~ **Initial Catalog:** **TEAMXX_TenantDataDb~~x~~** **Username:** **TEAMXX ~~SourceSQLuserX~~** **Password:** **TEAMXX** ~~SourceSQLpwdX~~ Click the "**Change Connection String"** button to apply the connection string modifications		Use the parameters from ~~your~~ the Appendix in the "Hands-on Lab - Data Migration~~Workshop Sheet – Parameters~~" document.~~-~~ The connection string will now have been set to connect to the legacy SQL host – **LEGACYSQL2008** with appropriate Team database and login details.

Formatted: Font: 12 pt, Font color: Red

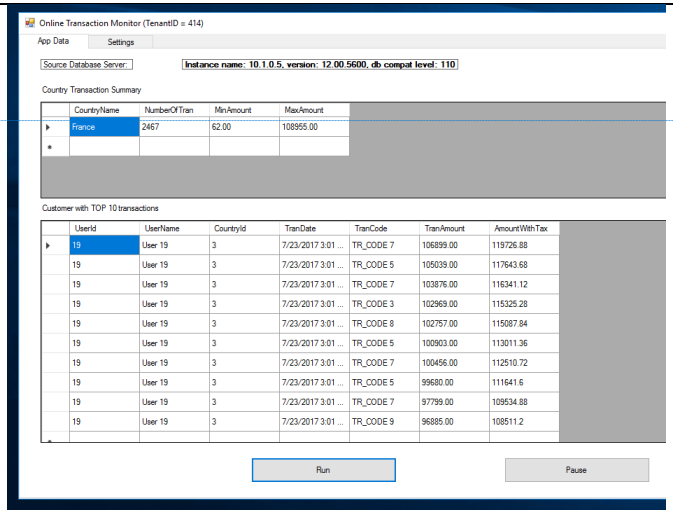
Formatted: Font color: Red

Formatted: Font: Bold

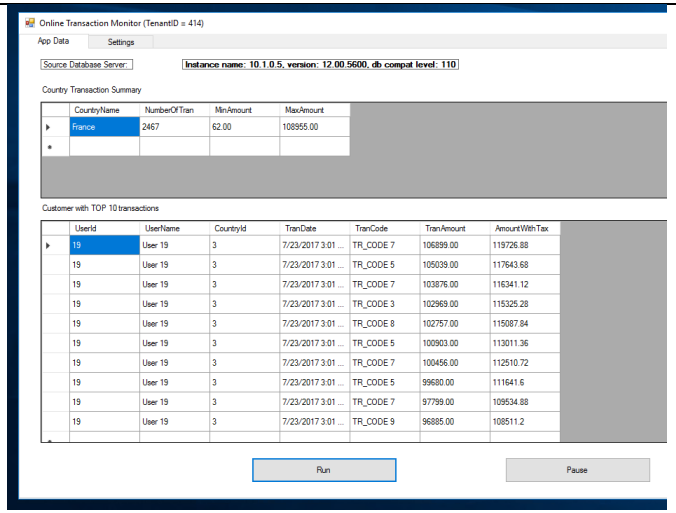
Formatted: Font: 12 pt

Formatted: Font: (Default) Calibri, 14 pt, Bold, Font color: Red

Formatted: Font: 12 pt

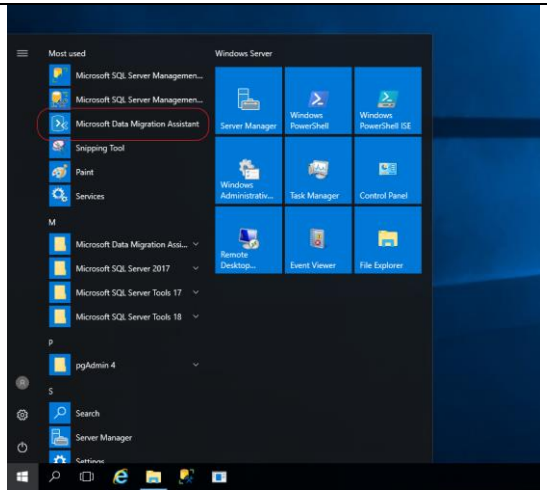
~~Select '**Change Connection String**'~~ ~~Se~~Select '**App Data**' tab and click the "**Run**" button.		The application will generate simulated transactional data. Notice how the 'Source Database Server' information at the top of the app reflects the parameters given in the previous step.~~The connection string will now have been set to connect to the legacy SQL host – SourceSQLhost~~

Formatted: Font: Bold, Font color: Blue

~~Select 'Run'~~		~~The application will generate simulated transactional data. Notice how the 'Source Database Server' connection reflects the parameters given in the previous step.~~

2. Assess the ~~A~~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. RDP onto **your teams Win10 Management VM (TEAM-VMXX)** ~~WMHx~~ and run DMA from the Start or Desktop icon.		See link above if you need to download DMA.

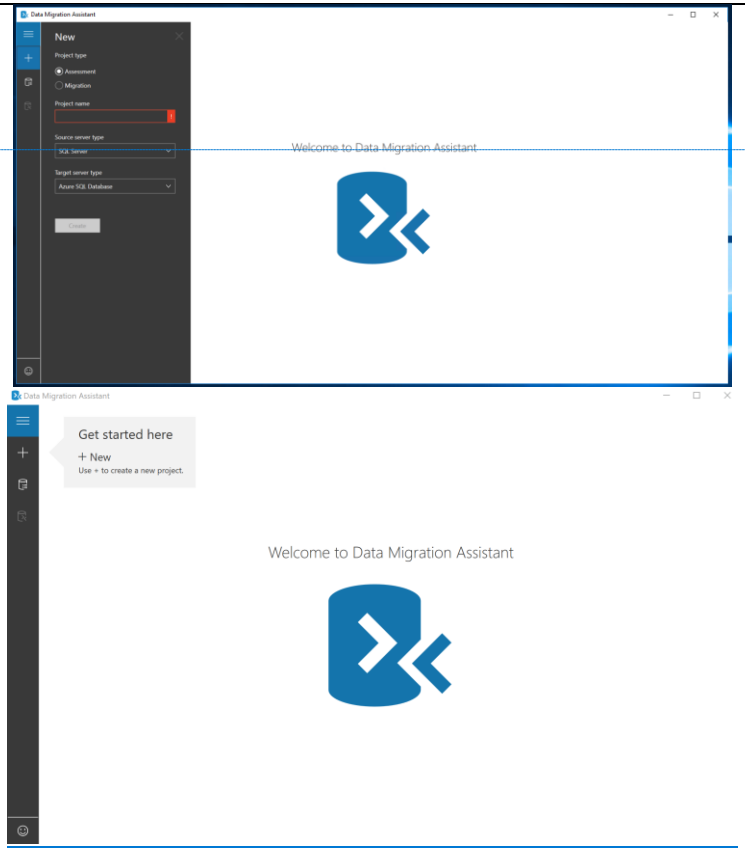
Formatted: Numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 0.63 cm + Indent at: 1.27 cm

Formatted Table

Formatted: Indent: Left: 0 cm, Hanging: 0.07 cm

You should see this screenshot to the right.

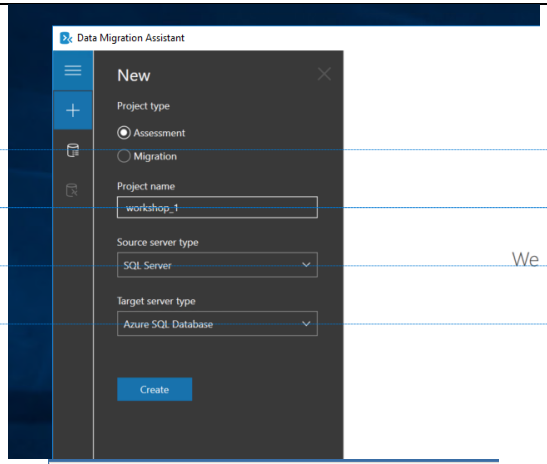
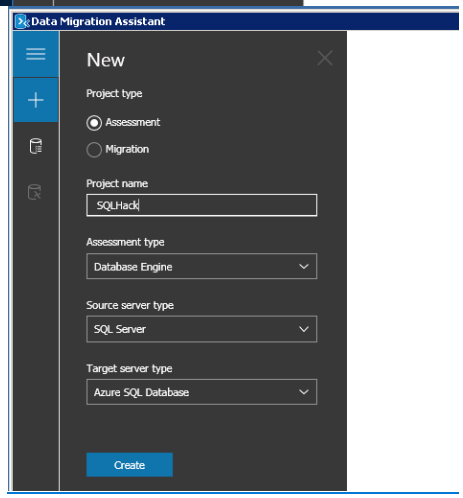
Select the **"+"** to create a new **assessment** project



Formatted: Font: Bold, No underline

Formatted: Font: Not Bold

Formatted: Centered

Select/Enter the following details: - ~~p~~**Project name**: **Workshop1** **Assessment type:** **Database Engine** - **Source server type:** **~~Select '~~ SQL Server~~'~~** - **~~Select '~~Target server type:** **Azure SQL Database~~'~~** ~~Select~~ Click **'Create'**	 We 	Our first project assessment assumes we will be migrating to Azure SQL DB, so the selections shown in the screenshot need to be selected.

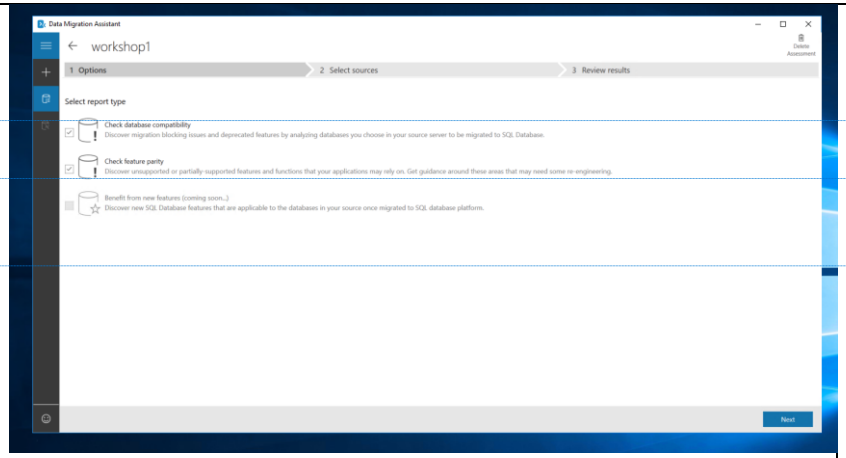
Formatted: Font: Bold

Formatted: Font: Bold, Font color: Red

Formatted: Font: Bold

Formatted: Font: Bold, Font color: Red

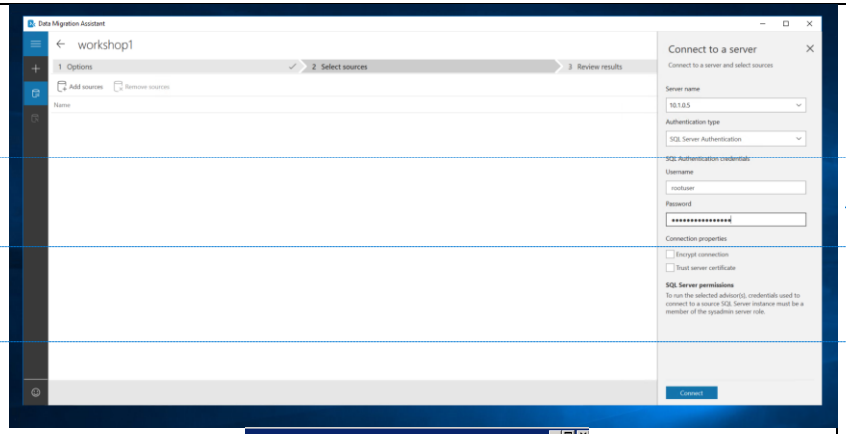
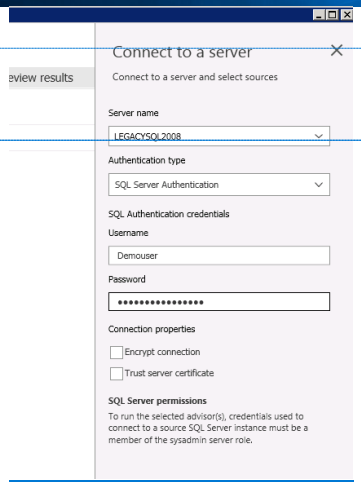
Formatted: Font: Bold, Font color: Red

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.

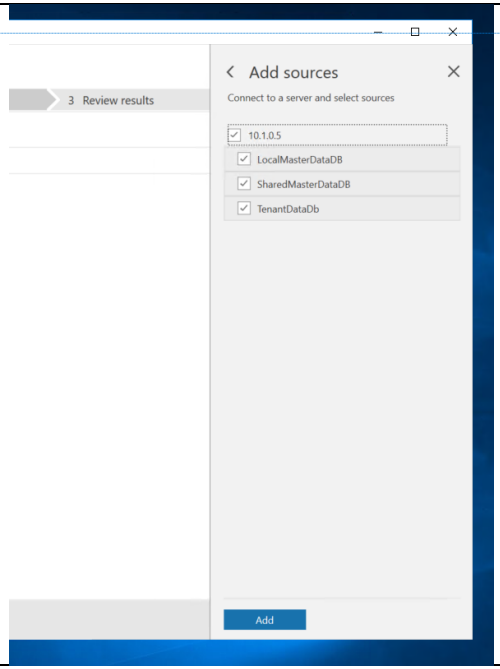
Formatted: Font: Bold

Formatted: Font: Bold

Formatted: Font: Bold

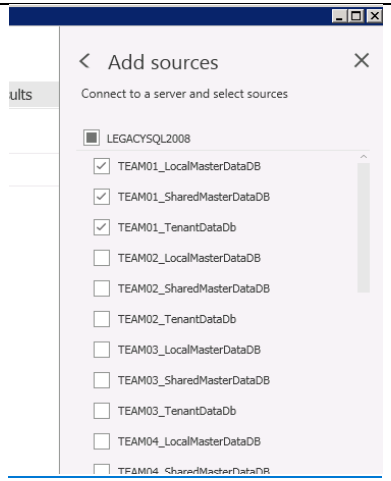
Enter the ~~S~~source/legacy SQL details: **Server Name:** ~~SourceSQLhostX~~10.0.2.4 **Authentication Type:** 'SQL Server Auth~~-~~entication **Username:** ~~SourceSQLuserX~~Demouser **Password:** ~~SourceSQLpwdX~~Demo@pass1234567 Untick "Encrypt connection" Click '**Connect**'	 	When performing this within your own subscription you will enter the host, authentication and connection types according to your company guidelines and practices. *Bear in mind that DMA needs to connect to a source SQL Server using an account that belongs to the sysadmin role.* As this document is produced within a workshop environment Active Directory, Certificates and encryption has not been setup.

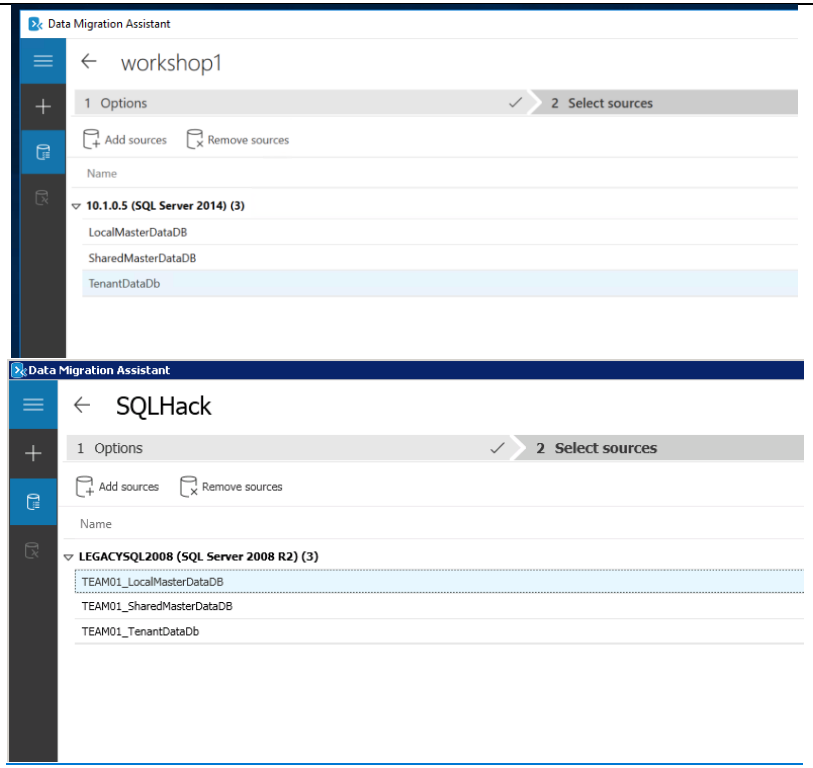
Formatted: Indent: Left: 0 cm
Formatted: Font: Bold
Formatted: Font: Italic, Font color: Red
Formatted: Font: Bold, Font color: Red
Formatted: Font: Bold
Formatted: Font: Italic, Font color: Red
Formatted: Indent: Left: 0 cm, First line: 0 cm
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Indent: Left: 0 cm
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Font color: Auto
Formatted: Indent: Left: 0 cm, First line: 0 cm
Formatted: Font color: Accent 1
Formatted: Font color: Auto
Formatted: Font: Bold, Font color: Auto
Formatted: Indent: Left: 0 cm

Select **only** the 3 database used by your ~~The three databases associated with the~~ 'Online Transaction Monitor' ~~are:~~app. These will have a TEAMXX prefix where XX should be replaced by your team number. **TEAMXX_LocalMasterDataDb**~~x~~ **TEAMXX_SharedMasterDb**~~x~~ **TEAMXX_TenantDataDb**~~x~~ ~~Select~~ Click '**Add**' to add them to the ~~evaluation~~assessment.		DMA will ~~evaluate~~ show all ~~the~~ database<u>s</u> 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.

Formatted: Font: Bold

Formatted: Indent: Left: 0 cm

	 Add sources Connect to a server and select sources ults LEGACYSQL2008 TEAM01_LocalMasterDataDB TEAM01_SharedMasterDataDB TEAM01_TenantDataDb TEAM02_LocalMasterDataDB TEAM02_SharedMasterDataDB TEAM02_TenantDataDb TEAM03_LocalMasterDataDB TEAM03_SharedMasterDataDB TEAM03_TenantDataDb TEAM04_LocalMasterDataDB	

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

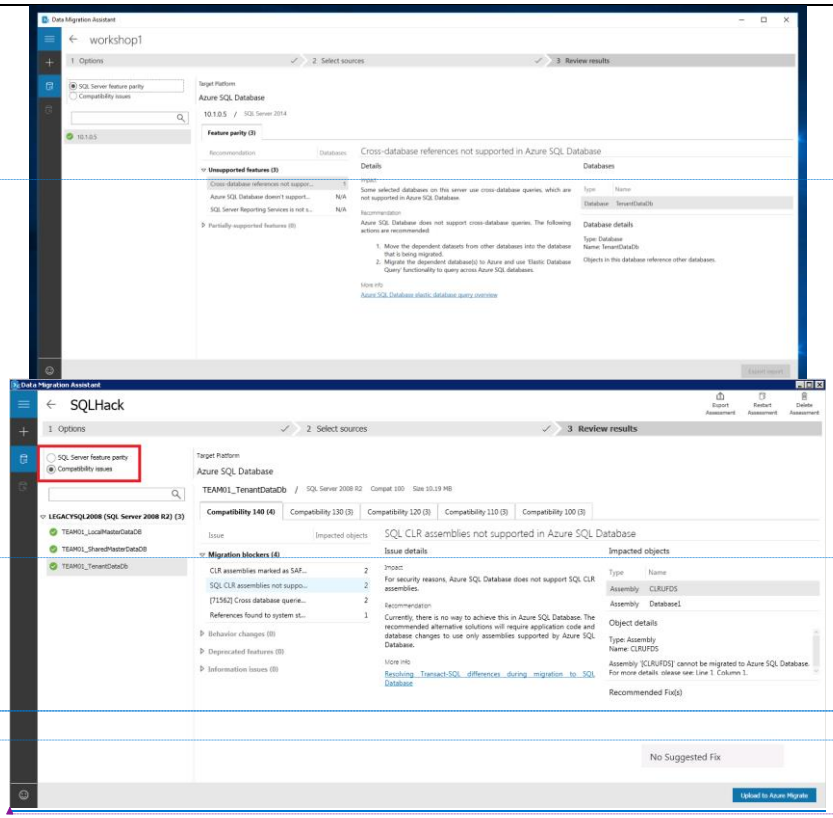
DMA will now show the results ~~from~~ of the ~~tests~~ assessment using 2 separate reports: ~~you selected earlier.~~

‘~~**Note:**~~

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

‘**Compatibility Issues**’ which is a database level report detailing individual objects that have compatibility issues.
Select
‘TEAMXX_TenantDataDb~~X~~’ ~~.~~
Note the 4 ‘Migration blockers’ including CLR which the ~~Notice that~~ database uses. ~~‘CLR’ is required for the application to run.~~

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



Note: Toggle the parity and compatibility ~~i~~Issues radio button (top left) to switch between the 2 reports. ~~see how DMA.~~

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

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

Formatted: Font: Bold

Field Code Changed

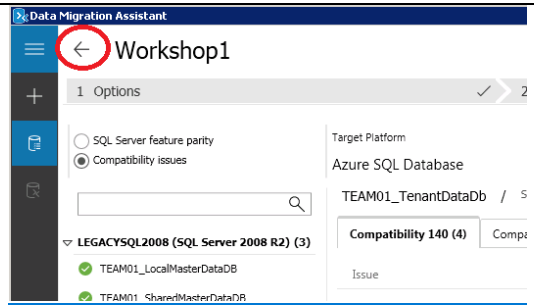
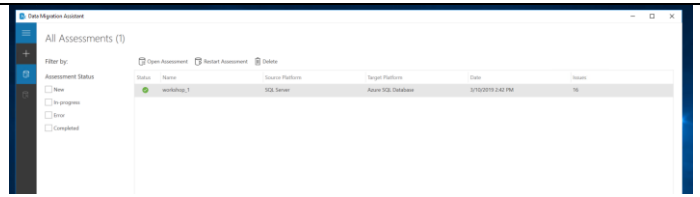
Formatted: Font: Bold

Formatted: Font: Bold, Font color: Red

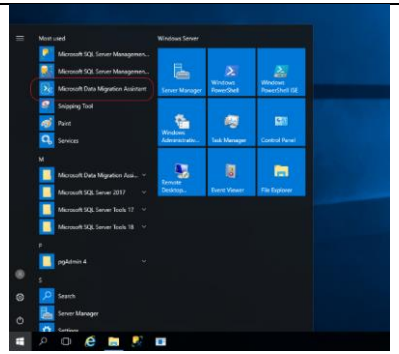
Formatted: Font: 11 pt, Bold

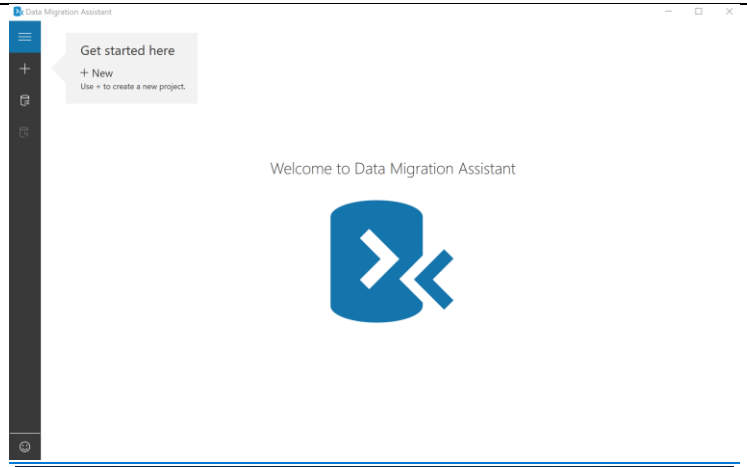
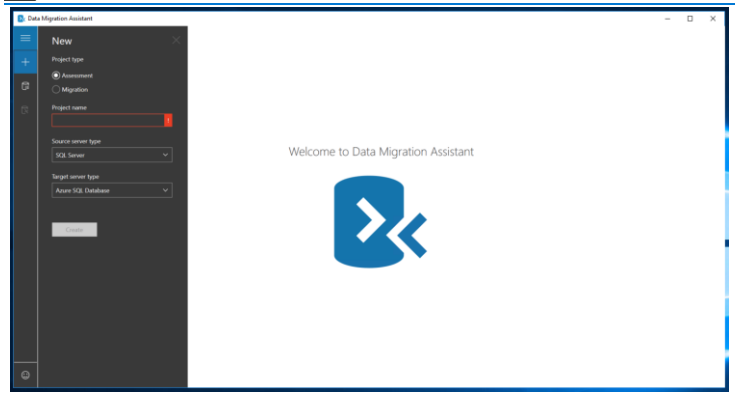
Formatted: Font color: Auto

Formatted: Font: 11 pt

~~Azure SQL Database Managed Instance (SQLMI) does support CLR. SQLMI is the Microsoft SQL **Instance** as a Service.~~		
Once you're reviewed the assessment click the back arrow to see a list of current DMA projects.		
		
	Due to the need for CLR, we need to repeat the assessment with Azure SQL DB Managed Instance as the target.	

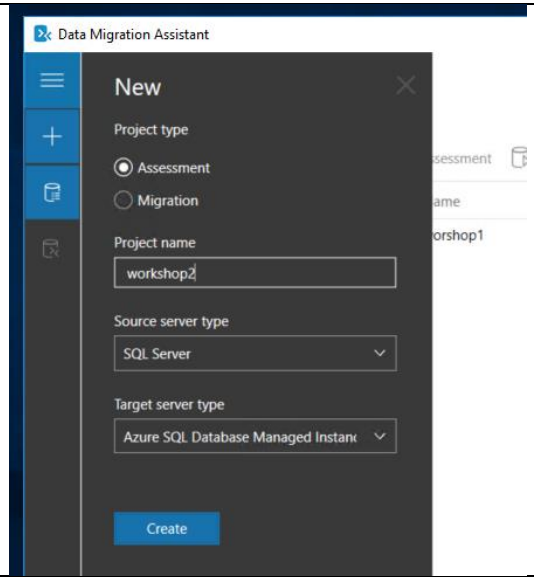
Formatted Table

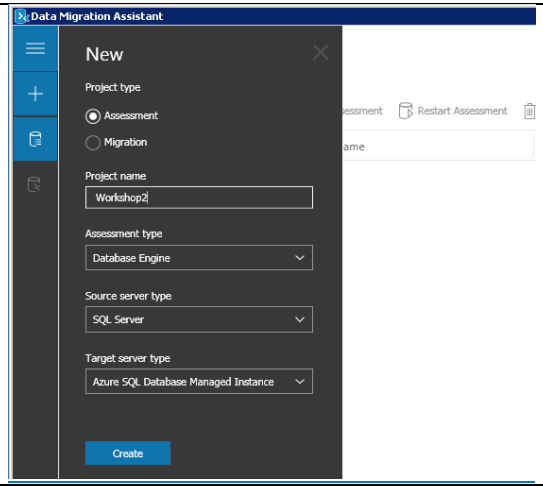
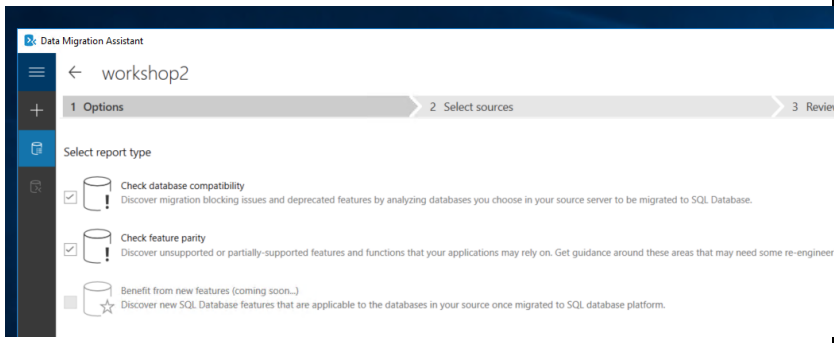
We need to determine the suitability of the database(s) for migration to Azure. This includes checking for compatibility and feature support with Azure Database. Logon onto your teams Win10 management VM ~~**WMHx**~~ and run DMA from the Start or Desktop icon.		See link above if you need to download DMA.

You should see this screenshot to the right. Select the **"+"** to create a new **assessment** project. ~~You should see this screenshot to the right.~~ ~~Select the "+" to create a new assessment~~	 	
Select/Enter the following details: **Project name:**		Our ~~first~~ 2nd ~~project~~ assessment project assumes we will be migrating to Azure SQL DB Managed Instance, so

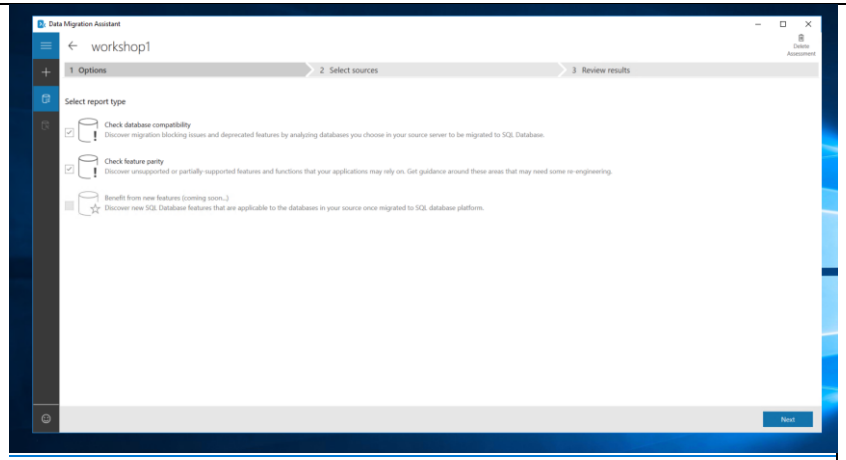
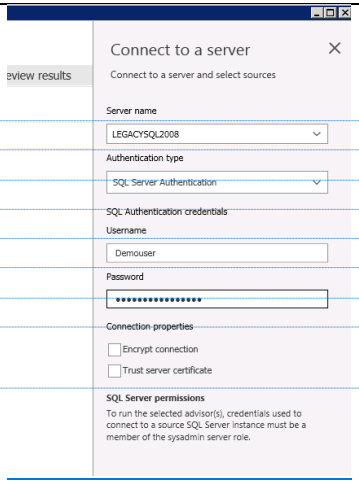
Formatted: Superscript

Formatted: Font: Bold

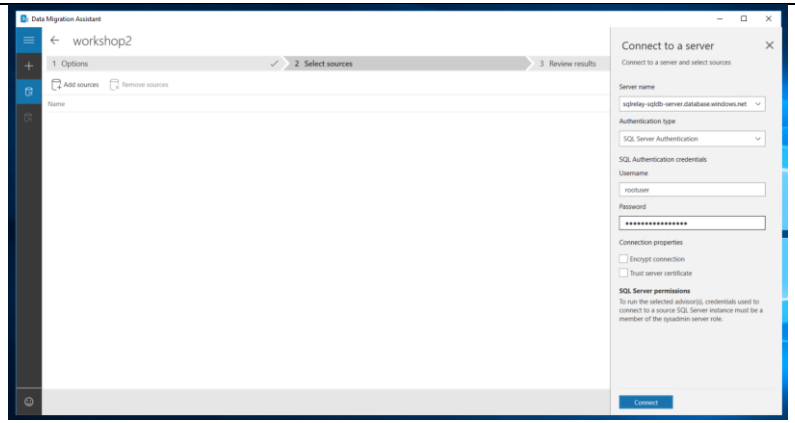
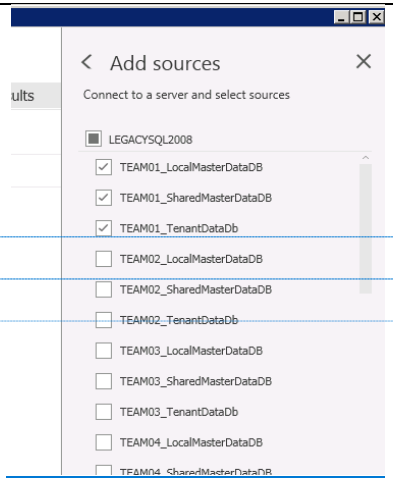
Workshop2 **Assessment type:** **Database Engine** **Source server type:** **SQL Server** **Target server type:** **Azure SQL Database Managed Instance** Click '**Create'** ~~= project name~~ ~~= Select 'SQL Server'~~ ~~= Select 'Azure SQL Database Managed Instance'~~ ~~Select '**Create'**~~			the selections shown in the screenshot need to be selected.

		
~~Select the checks (Report Type) to be made.~~		~~DMA can test for both database compatibility and feature parity compliance against the Azure target.~~ ~~As this is the first time we are assessing a database(s) we will perform all of these tests.~~

Formatted: Centered

Select the assessment checks (Report Type) to be made: **Check database compatibility** **Check feature parity** Click '**Next**'		As we saw previously DMA can test for both database compatibility and feature parity compliance against the chosen target. As before we will assess all the databases against all of the tests.
Enter the source/legacy SQL details: **Server Name:** **LEGACYSQL2008** **Authentication Type:** **SQL Server Authentication** **Username:** **Demouser** **Password:** **Demo@pass1234567** **Untick "Encrypt connection"** Click '**Connect**'		When performing this within your own subscription you will enter the host, authentication and connection types according to your company guidelines and practices. *Bear in mind that DMA needs to connect to a source SQL Server using an account that belongs to the sysadmin role.* As this document is produced within a workshop environment Active Directory, Certificates and encryption has not been setup.

Formatted: Font: Bold
Formatted: Font color: Accent 1
Formatted: Font: Bold
Formatted: Font color: Accent 1
Formatted: Font: Bold
Formatted: Font color: Accent 1
Formatted: Font: Bold
Formatted: Font color: Accent 1
Formatted: Font color: Accent 1

~~Enter the Source SQL details:~~ ~~Server Name~~ ~~SourceSQLhost✘~~ ~~Authentication Type~~ ~~'SQL Auth.'~~ ~~Username~~ ~~SourceSQLuser✘~~ ~~Password~~ ~~SourceSQLpwd✘~~		~~When performing this within your own subscription you will enter the host, authentication and connection types according to your company guidelines and practices. As this document is produced within a workshop environment Active Directory, Certificates and encryption has not been setup.~~
Select **only** the 3 database used by your 'Online Transaction Monitor' app. These will have a TEAMXX prefix where XX should be replaced by your team number. **TEAMXX_LocalMasterDataDb** **TEAMXX_SharedMasterDb** **TEAMXX_TenantDataDb** Click '**Add**' to add them to the assessment.		DMA will show all databases located on the Source host and display them so you can decide which ones to include in this assessment project. Note that you can assess multiple databases a t the same time.

Formatted Table

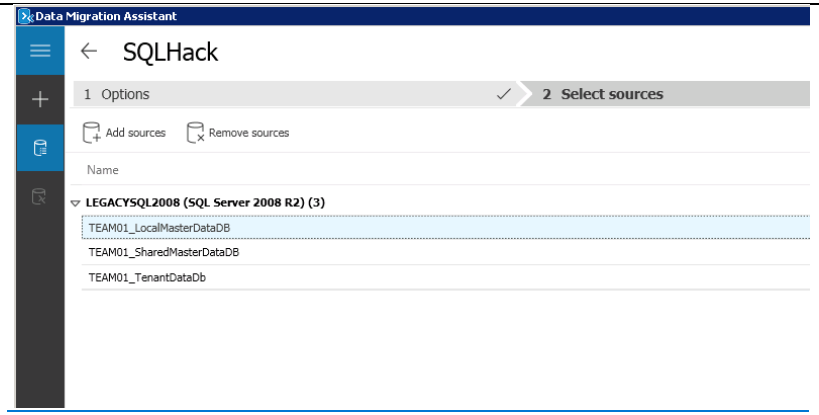
Formatted: Font: 11 pt, Bold

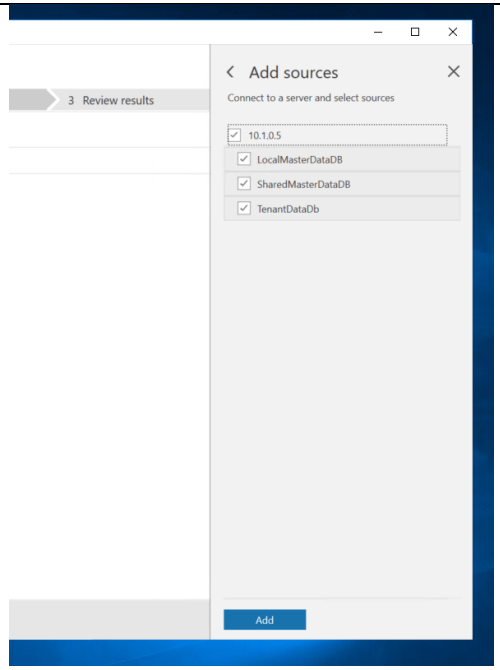
Formatted: Font: 12 pt

Formatted: Font: 11 pt, Bold

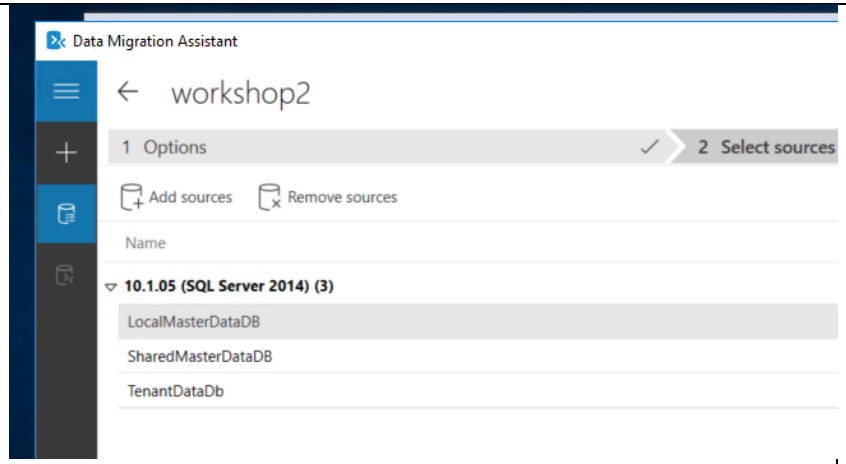
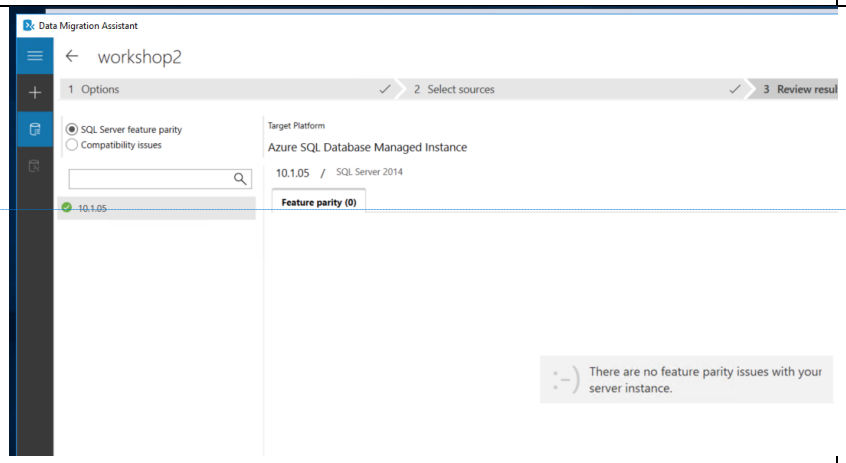
Formatted: Font: 12 pt

Formatted: Font: 11 pt, Bold

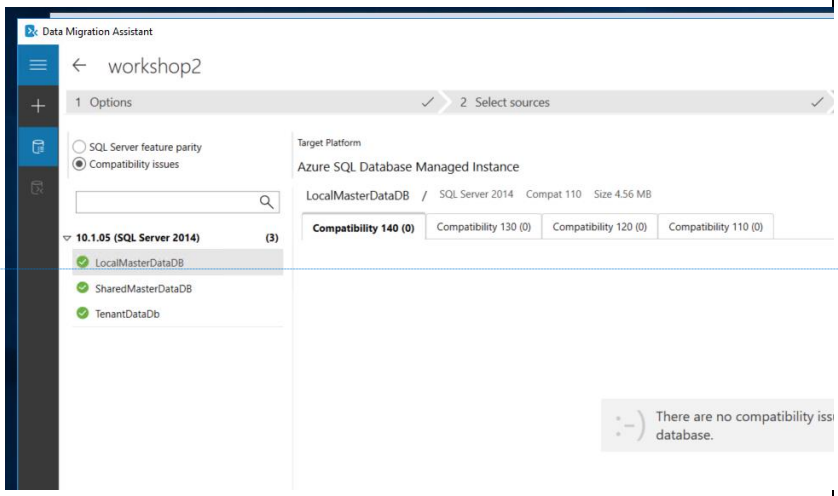
You should now see the screen on the right with the relevant TEAMXX databases listed. Select '**Start Assessment**'		Note: DMA allows you to either 'Add' or 'Remove' additional data sources as needed at this point. Also note that DMA has identified what compatibility level each source database is running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

~~The three databases associated with the 'Online Transaction Monitor' are:~~ ~~LocalMasterDataDb~~**~~x~~** ~~SharedMasterDb~~**~~x~~** ~~TenantDataDb~~**~~x~~** ~~Select '**Add**' to add them to the evaluation.~~		~~DMA will evaluate the database located on the Source host and display them.~~

Formatted Table

~~You should now see the screen on the right, or the equivalent datasources that you have selected.~~ ~~Select '**Start Assessment'~~		~~Note: DMA allows you to either 'Add' or 'Remove' additional datasources as needed at this point.~~
As before DMA will now show the results from the assessment as the 2 reports.~~tests you selected earlier.~~ Note the ~~:~~ '**SQL Server feature parity**' report will either be clean or it will show a single issue for the system SQL Agent Job 'syspolicy_purge_history' which is not applicable to Azure SQL DB Managed Instance & can be ignored. ~~– is clear of issues.~~		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

Formatted: Font: Bold

The '**Compatibility Issues**' report should be clear for all 3 databases showing that they can be migrated to Azure SQLDB Managed Instance without changes. ~~– is clear of issues.~~ ~~Azure SQL Database Managed Instance (SQLMI) does support CLR. SQLMI is the Microsoft SQL **Instance** as a Service (PaaS).~~		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 can now migrate the application to Azure SQL Database Managed Instance.~~	**We are now ready to migrate the application databases to Azure SQL Database Managed Instance**	

Formatted: Font: Bold

Formatted: Centered

3. Use Azure Database Migration Service (DMS) to m~~M~~igrate the 3 application ~~D~~databases

Narrative	Screenshot	Notes

Formatted: Numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 0.63 cm + Indent at: 1.27 cm

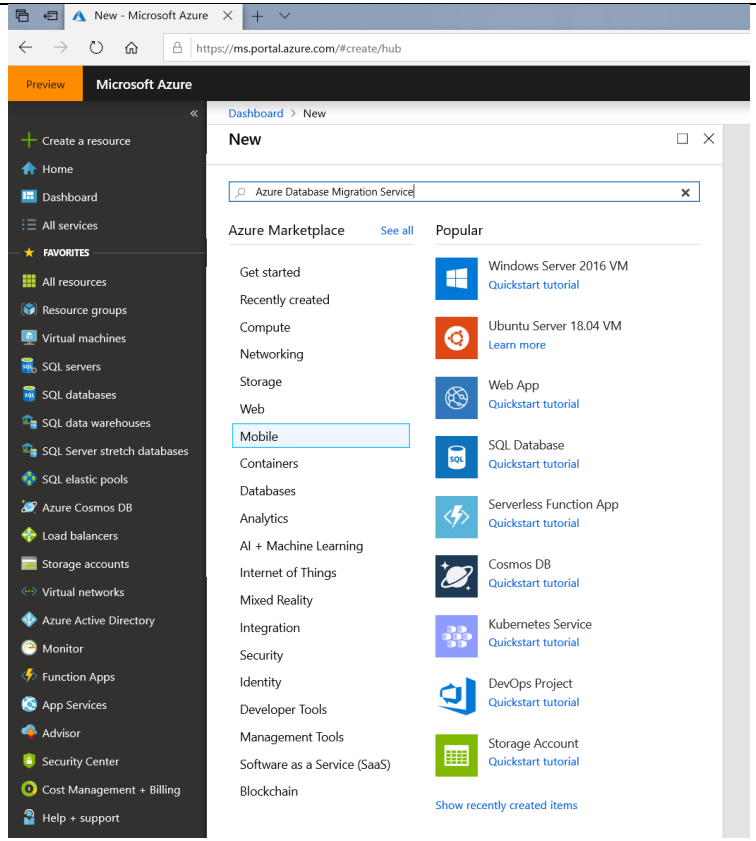
Formatted Table

We will be using Azure Database Migration Service (DMS) to migrate the legacy databases to Azure.

For the workshop DMS will already been provisioned. ~~configured.~~ In your own subscription if you do not have DMS:

- select '**+ Create a Resource**'
- enter and search for 'Database Migration'
- select '**Azure Database Migration Service**'

Follow the 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 SQL Managed Instance.
DMS can host and run different types of database migration Projects under the same e.g. separate Project for separate source servers.

Overview of DMS: Azure Database Migration Service ~~(DMS)~~

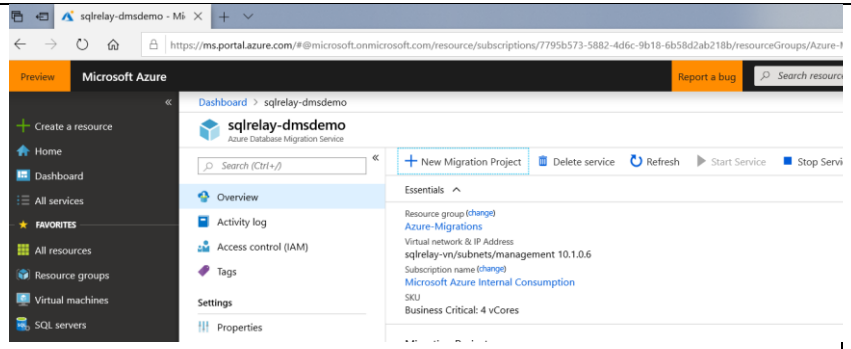
DMS tutorials: https://docs.microsoft.com/en-us/azure/dms/

Formatted: Indent: Left: 0.15 cm, Hanging: 0.39 cm

For this workshop:-

In the Azure portal, open the workshop Resource Group and locate the Azure Database Migration Service and open it.
~~= Select the DMS Service~~

On the DMS Overview blade-
~~= Select~~ click **'New Migration Project'**



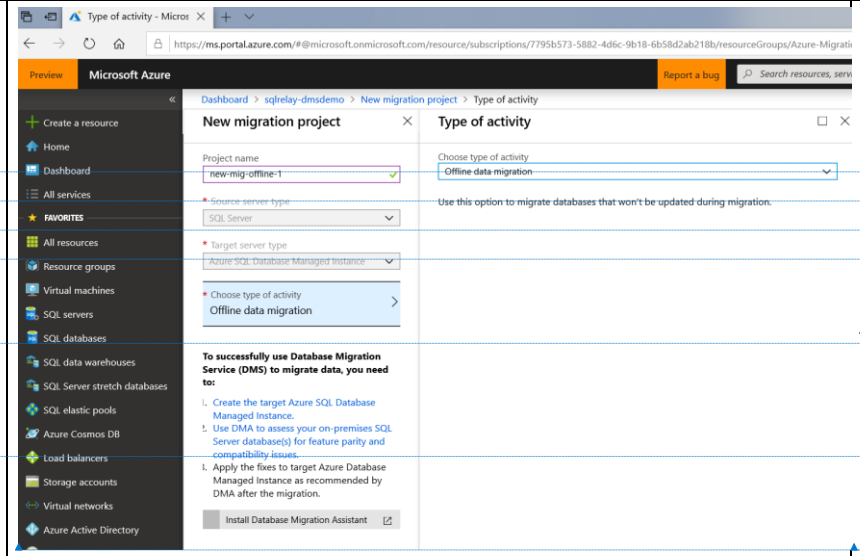
On the "New migration project" blade use the following settings:-

~~= Enter the~~ **'Project name:'**
TEAMXX_migration
(replace XX with your team number)

Source server type:
~~= Select~~ **'SQL Server'**

~~= Select~~ **Target Server type:**
'Azure SQL ~~=~~ Database Managed Instance'

Choose type of activity: ~~= Select~~
'Offline data migration'



~~We will be running an 'Offline' DMS migration in this workshop.~~ DMS can perform two types of database migrations:

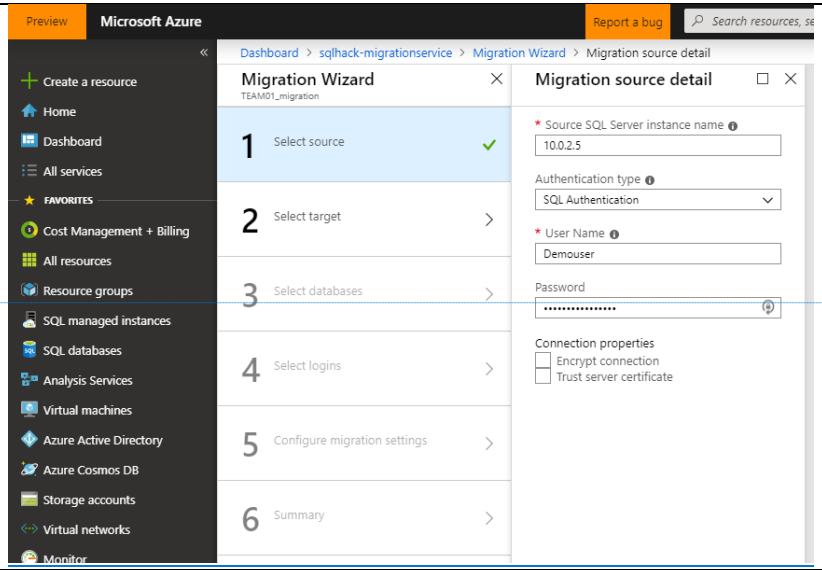
- ~~-~~ **Offline**
- ~~- Online~~
-
- **Onffline**

Offline migrations ~~are~~ use ~~migrations from~~ backup files. The backups can be provided to DMS or DMS can ~~also~~ 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.

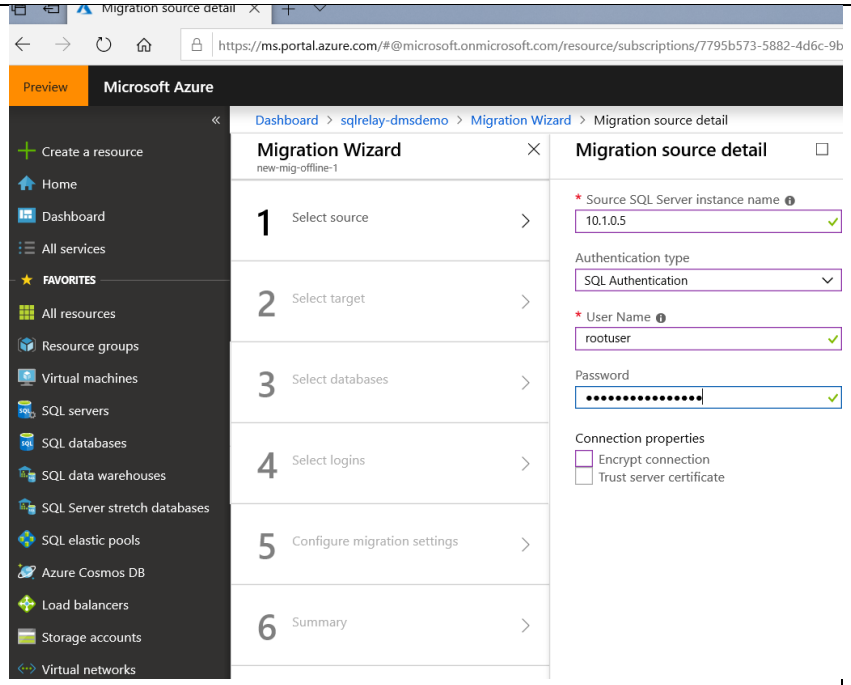
Formatted: Font: Bold, Font color: Accent 1
Formatted: Font: Bold, Font color: Accent 1, Check spelling and grammar
Formatted: Space Before: 6 pt
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: List Paragraph, Indent: Left: 0.18 cm, Hanging: 0.39 cm, Bulleted + Level: 1 + Aligned at: 0.63 cm + Indent at: 1.27 cm
Formatted: Font: Bold, Font color: Red
Formatted: Font: Italic
Formatted: Indent: Left: 0.18 cm, Hanging: 0.39 cm
Formatted: Font: Italic
Formatted: List Paragraph, Indent: Left: 0.18 cm, Hanging: 0.39 cm, Bulleted + Level: 1 + Aligned at: 0.63 cm + Indent at: 1.27 cm
Formatted: Space Before: 6 pt
Formatted: Font: Bold
Formatted: Space Before: 6 pt
Formatted: Font: Bold
Formatted: Space Before: 6 pt
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Space Before: 6 pt
Formatted: Centered

~~Select~~ Click **'Save'** Select '**Create and run activity**'		**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 ~~start~~ launch the migration configuration blades.~~the migration activity.~~ Use the following values for each of the configuration steps:~~We need to supply the parameters required for this activity.~~ **STEP 1: Select ~~Migration~~ Source ~~Detail~~** *This ~~is~~ uses the source database host VM LEGACYSQL2008 ~~/server~~ details ~~.~~ ~~F~~from the "~~Migration Workshop~~ Lab ~~–~~and parameters" doc.~~:~~ ~~Enter:~~**Source SQL Server: 10.0.2.4** ~~= SourceSQLhost~~**~~X~~** **Authentication Type:** ~~=~~ **SQL Authentication** **User Name:**		The account that DMS uses to connect to the source instance must be a member of sysadmin.

Formatted: Font: Bold, Font color: Accent 1
Formatted: Space Before: 6 pt
Formatted: Font: Italic
Formatted: Font: Bold
Formatted: Font: Italic, Font color: Red
Formatted: Font color: Auto
Formatted: Space Before: 6 pt
Formatted: Font: Bold, Font color: Red
Formatted: Font: Bold
Formatted: Space Before: 6 pt

Demouser Password Demo@pass1234567 ~~= SourceSQLuser~~ ~~= SourceSQLpwd~~ For this lab **only** uncheck both “Connection Properties” options as per the ~~diagram~~screenshot. ~~Select~~ Click ‘**Save**’. This will perform a connection test.		

Formatted: Font: Bold


Migration source detail
https://ms.portal.azure.com/#@microsoft.onmicrosoft.com/resource/subscriptions/7795b573-5882-4d6c-9b
Preview
Microsoft Azure
Dashboard > sqlrelay-dmsdemo > Migration Wizard > Migration source detail
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
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 source detail
* Source SQL Server instance name
10.1.0.5
Authentication type
SQL Authentication
* User Name
rootuser
Password
Connection properties
Encrypt connection
Trust server certificate

STEP 2: Select Target

This uses the target Azure SQL Managed Instance details from the "Lab and parameters" doc. ~~**Migration Target Detail**~~

~~This is the target database host/server details. From the "Migration Workshop – parameters":~~

Target SQL Server:
(Use the Fully Qualified Domain Name for the SQL Managed Instance)
SqlhackmiXXXXXX
See C:_SQLHACK_\LABS\01-Data_Migration\ManagedInstanceFDQN.txt

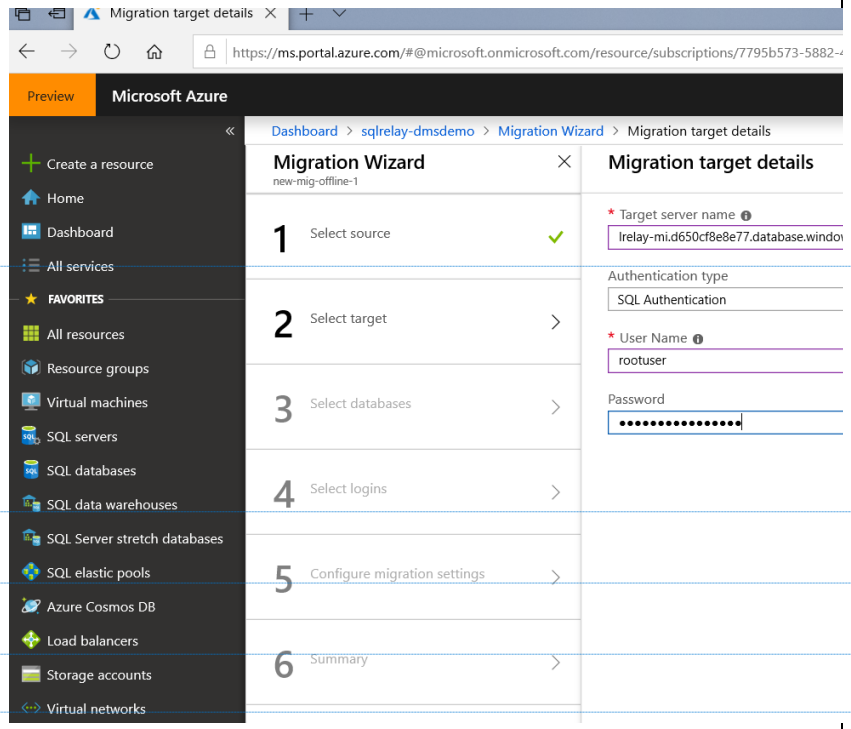
Authentication Type:
SQL Authentication

User Name:
Demouser

Password:
Demo@pass1234567

~~**Enter:**~~
~~= TargetMIhost~~

~~= SQL Authentication~~

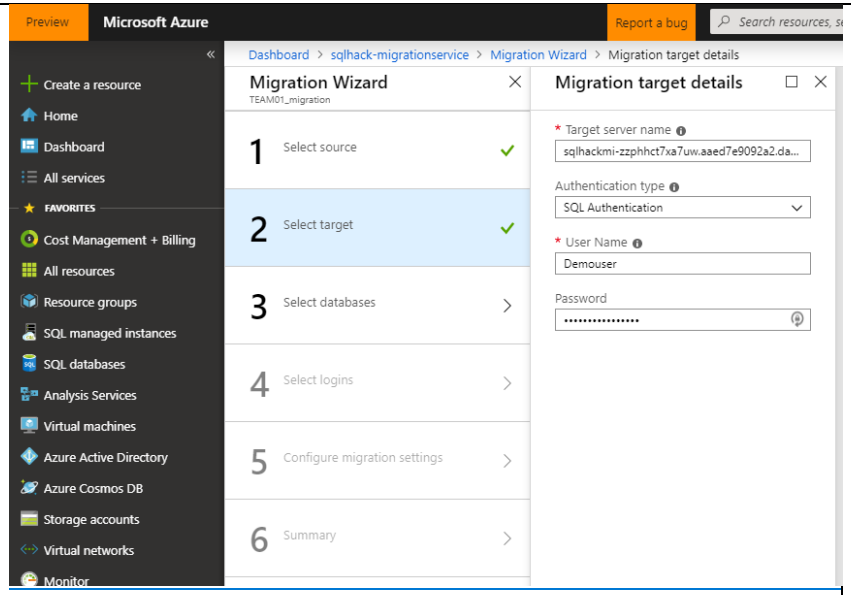


Formatted: Font: Bold
Formatted: Space Before: 6 pt
Formatted: Font: Bold
Formatted: Space Before: 6 pt
Formatted: Font: Bold
Formatted: Space Before: 6 pt
Formatted: Font: Bold
Formatted: Space Before: 6 pt
Formatted: Font: Bold

Click '**Save**'.

~~= TargetMIpwd**x**~~

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



STEP 3:

Select Databases

The application has 3 databases supporting it.

~~The workshop application utilises three databases. Ensure all three databases are selected.~~Select the 3 database for you team.

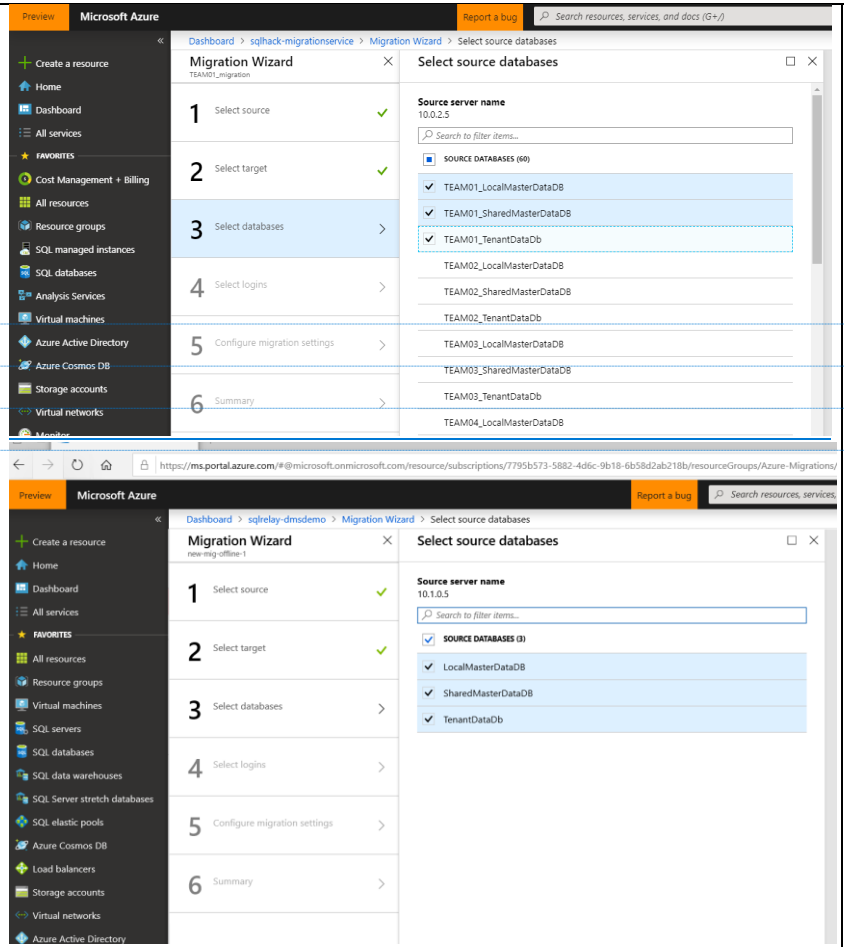
TEAMXX_LocalMasterDataDb

TEAMXX_SharedMasterDb

TEAMXX_TenantDataDb

(replace XX with your team number)

Select '**Save**'



Formatted: Font: Bold

Formatted: Font: Bold

Formatted: Font: Bold

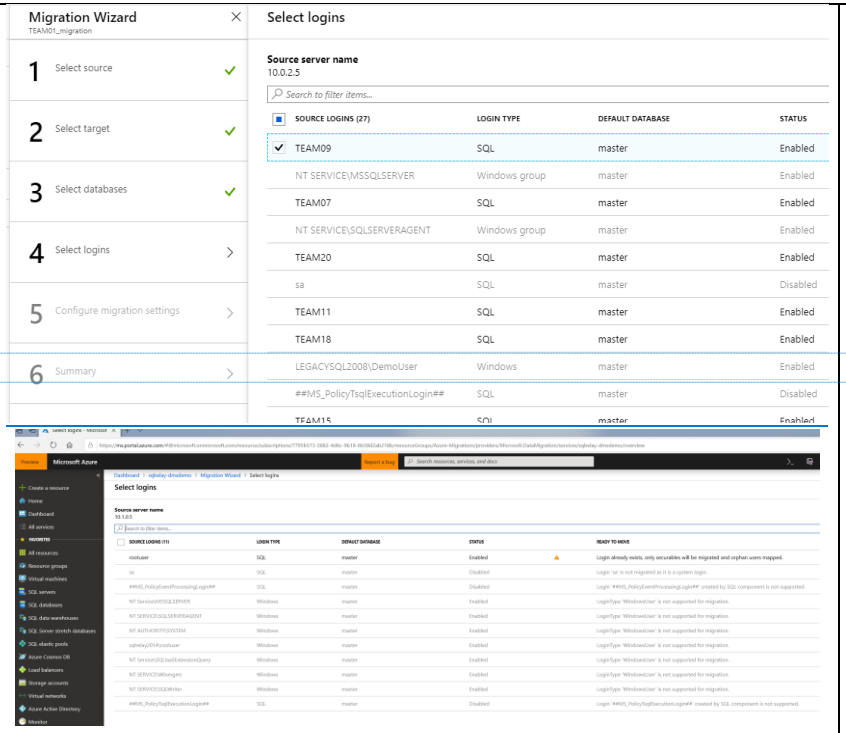
Formatted: Font: Italic

STEP 4:
Select Logins

As with a traditional on-premise migration 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 '~~SourceSQLuserx~~'TEAMXX' account.

Select 'Save'



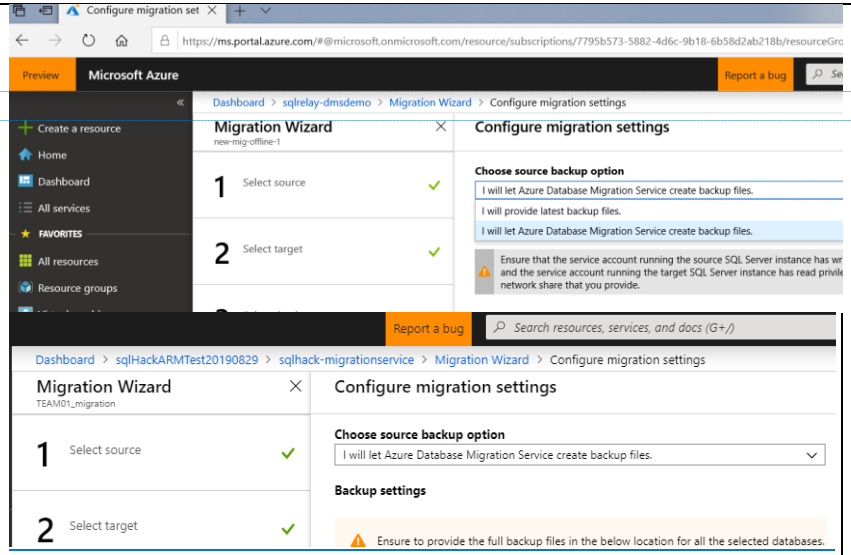
Formatted: Font: Bold

Formatted: Font color: Red

Step 5~~a~~
: Configure migration Settings
(Source Backup Option)~~(backups)~~

We are running an offline migration which will use backups of the DBs to be migrated. ~~DMS can either use backup files provided or take the backups as part of the migration activity.~~

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



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

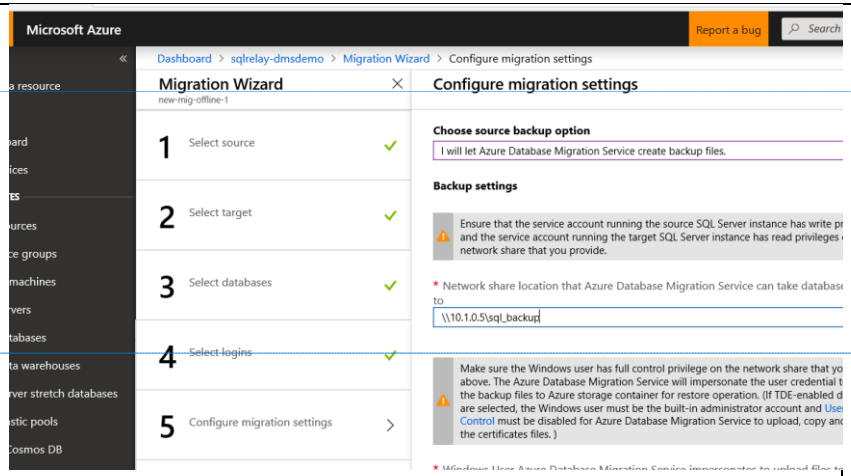
Formatted: Font: Italic

Formatted: Font: Italic

Step 5b:
Configure migration Settings *(backup location)*

We can now enter the Windows share that the source will write the database backups to. This is a Windows share which we'll enter in the format: ~~(i.e.: \\hostname\sql_backps").~~

\\IPAddress\FILESHARE

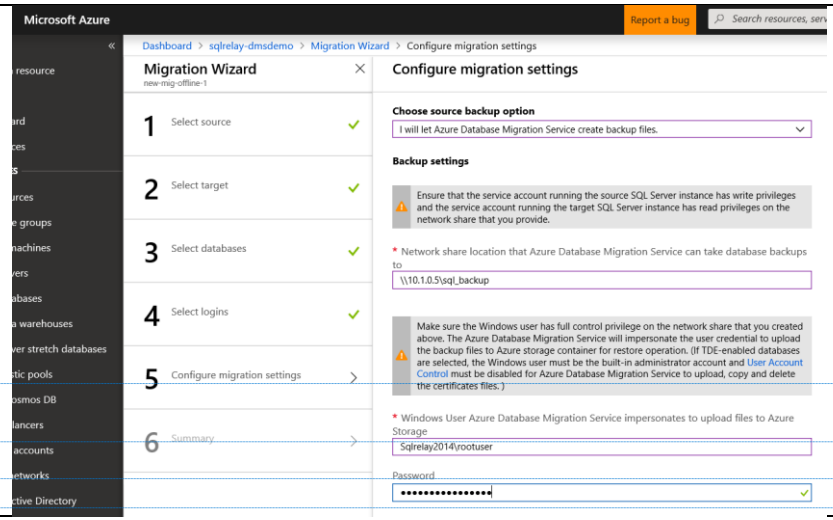


Step 5c:
Configure migration Settings (source database access)

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\~~SourceSQLuse~~ ~~r~~Demouser
Password:



Formatted: Font: Italic

Formatted: Font: Bold, Font color: Red

Formatted: Font: Bold, Font color: Red

Formatted: Font: Bold

Formatted: Font: Bold

Formatted: Font: Bold

Formatted: Font color: Auto

~~SourceSQLpwdX~~Demo@pass1234567		

Formatted: Font: Bold

Formatted: Font: Bold

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

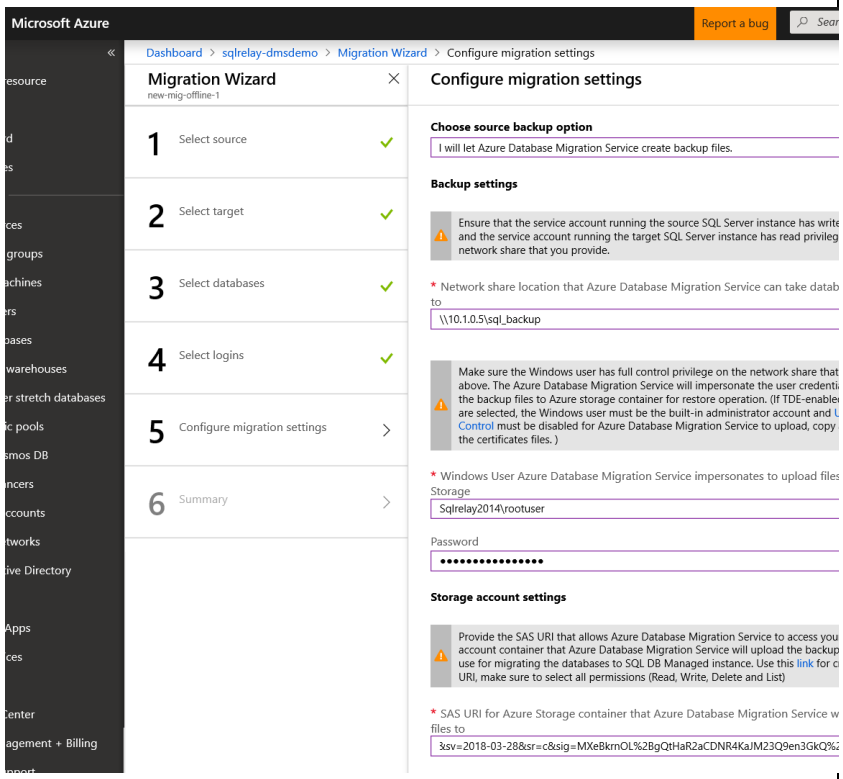
DMS is an Azure Cloud Service. We have to provide the Shared Access Signature ~~AS~~ URI (~~key~~SAS URI for short) to permit DMS to upload the backup files, from the share on the LEGACYSQL2008 host ~~in "5c"~~ to ~~an~~ Azure blob storage. ~~, for subsequent loading into the target.~~

The SAS URI is provided in the "Lab and Parameters" doc.

Enter the SAS URI key located in:
C:_SQLHACK_\LABS\01-Data Migration\SASKey.txt

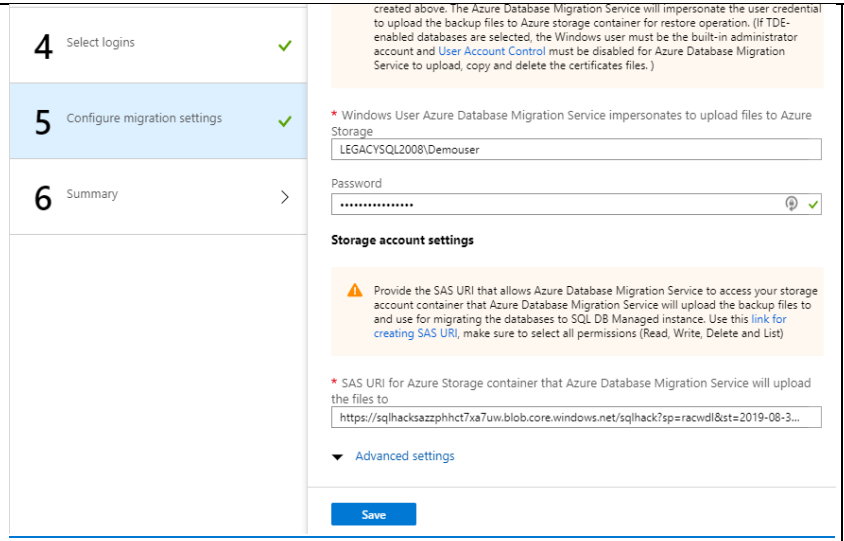
and click

~~Select~~ '**Save**'.
This will perform a connection test and if successful will display the Migration Summary blade.



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.

Formatted: Font: Italic


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://sqlhacksazzphhct7xa7uw.blob.core.windows.net/sqlhack?sp=racwdl&st=2019-08-3...
Advanced settings
Save

STEP 5e:

Configure migration settings *(Migration Summary)*

DMS displays the migration configuration settings ~~settings~~.

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

Use the following values:

~~Enter the migration activity name under "~~

Activity Name~~".~~

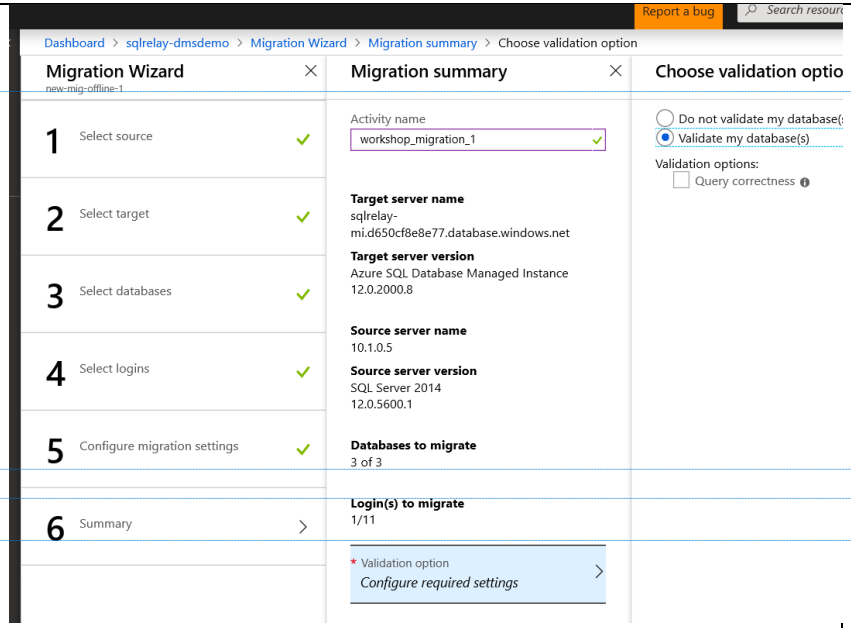
workshop_migration_1

Validation option:

~~Select '**Validation option**'~~

~~Select '~~ **Validate my database(s)**'

Select '**Save**'



Formatted: Font: Italic

Formatted: Font: Bold

Formatted: Font: Bold, Font color: Red

Formatted: Font: Bold

Formatted: Space Before: 6 pt

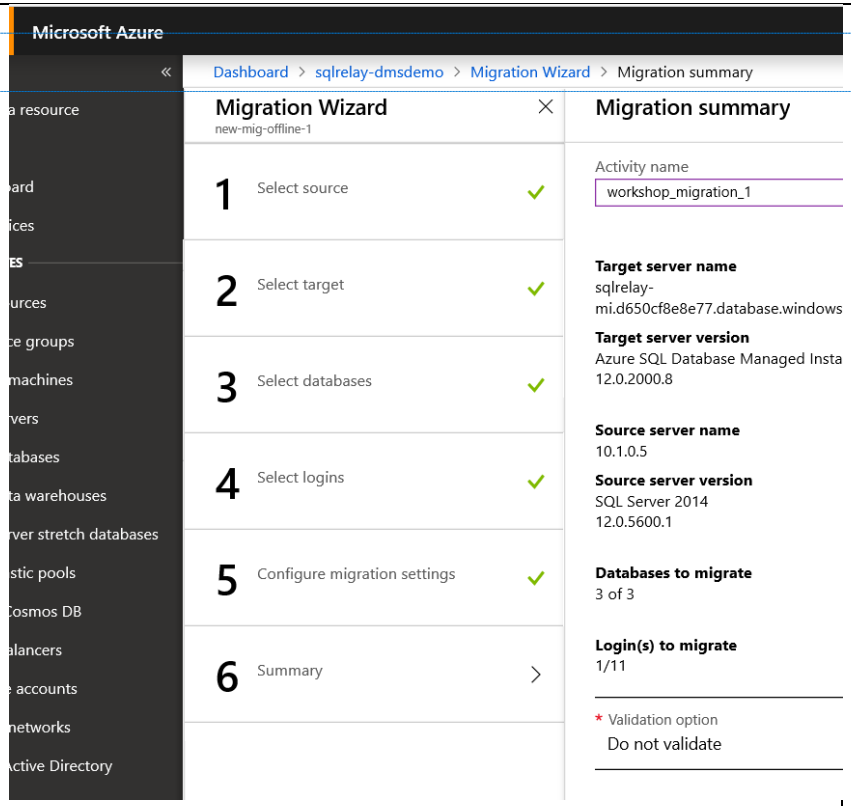
STEP 5f:

Migration Summary

(run the migration)

The migration activity is now configured and saved and is ready to run.

Select '**Run migration**'



Formatted: Font: Bold

Formatted: Font: Bold

Formatted: Font: Italic

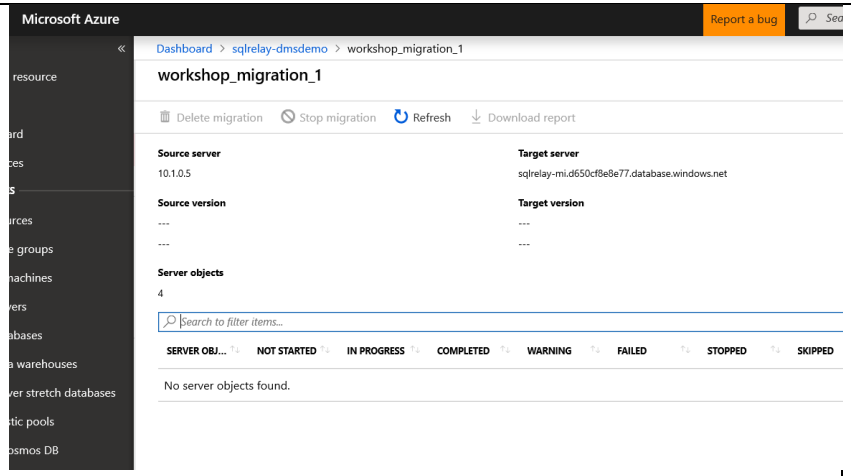
Formatted: Font: Bold, Italic

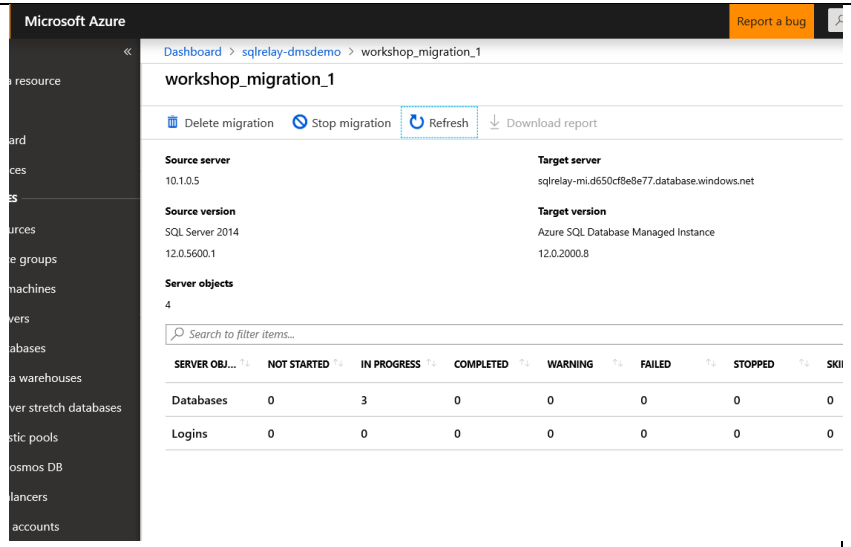
DMS will now run the migration activity.

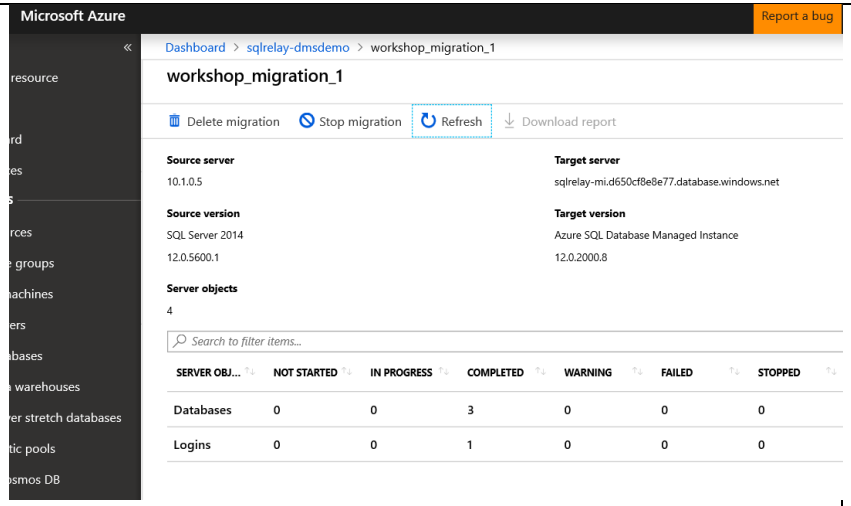
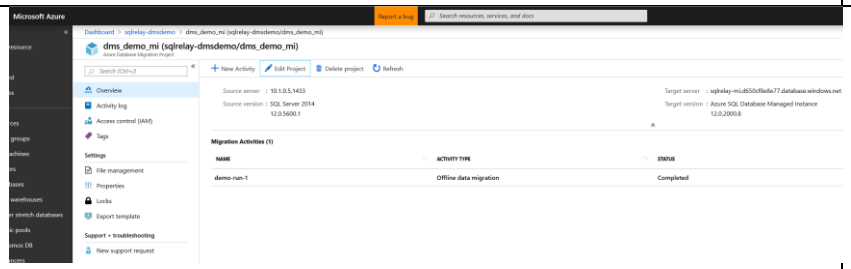
Initially this screen will be displayed.

Select '**Refresh**' to monitor the progress of your migration.

Notice the database counts under the following columns as you keep selecting '**Refresh**':
- "IN PROGRESS"
- "COMPLETED"
- "FAILED"




Microsoft Azure
Report a bug
Dashboard > sqlrelay-dmsdemo > workshop_migration_1
workshop_migration_1
Delete migration
Stop migration
Refresh
Download report
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
Search to filter items...
SERVER OBJ...
NOT STARTED
IN PROGRESS
COMPLETED
WARNING
FAILED
STOPPED
Databases 0 3 0 0 0 0 0
Logins 0 0 0 0 0 0 0

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.

1.4. Confirm application databases have been migrated & configure

We can also validate the migration by using SSMS.

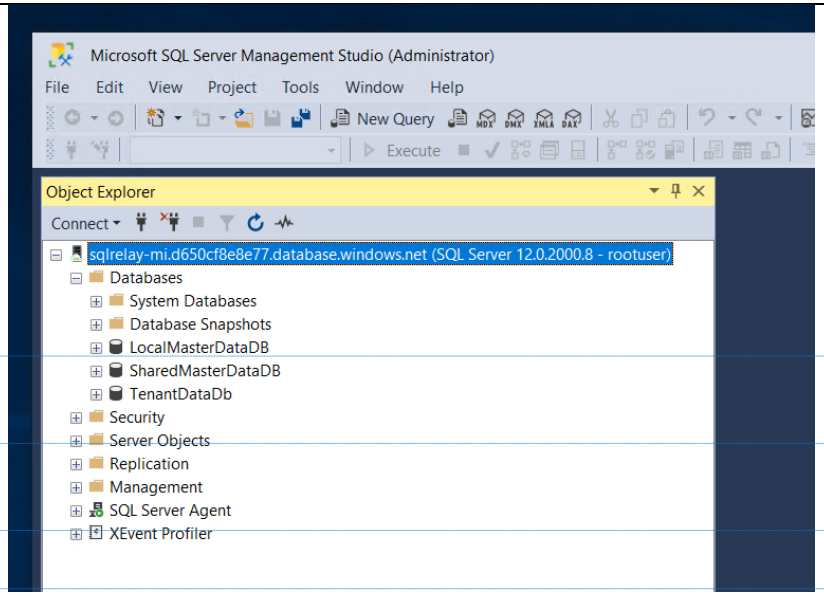
Log ~~back~~ into your teams ~~the~~ Windows 10 management VM ~~[WMHx].~~ ~~o~~Open ~~SSMS~~ SQL Management Studio and connect to the target Azure SQL Database Managed Instance using these details:~~.~~

Server:
(Use the Fully Qualified Domain Name for the SQL Managed Instance)
SQL Authentication
Username:
TEAMXX
Password:
TEAMXX

~~TargetMIhost~~
~~TargetMIuser~~
~~TargetMIpwd~~

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

TEAMXX_LocalMasterDataDb



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

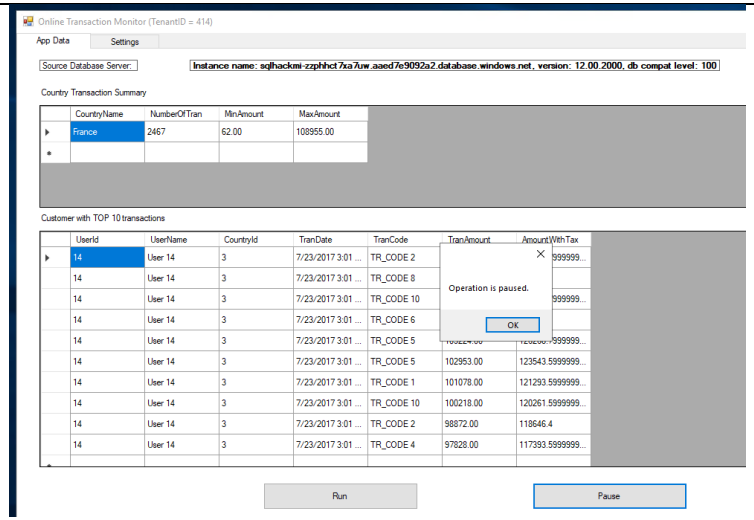
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Indent: Left: 0 cm
Formatted: Font: 11 pt, Bold, Font color: Auto
Formatted: Font: 11 pt, Font color: Auto
Formatted: Indent: Left: 0 cm

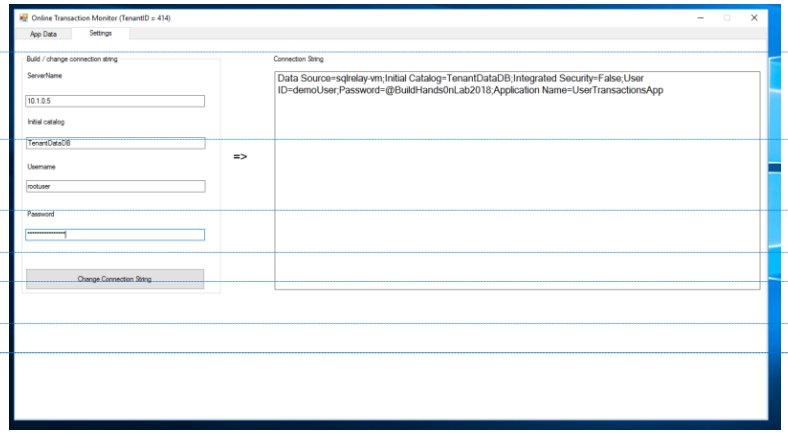
TEAMXX SharedMasterDb TEAMXX TenantDataDb		
Before we can test the application will run with the newly migrated databases we need enable CLR on the SQL Managed Instance. In SQL Server Management Studio open a new query window. *Make sure that you use the sysadmin account (below) for this new query window* **Sysadmin account:** **Username:** **Demouser** **Password:** **Demo@pass1234567** Once the query window has opened enter and run the TSQL statements opposite to enable CLR in the SQL Managed Instance.	sp_configure 'clr enabled',1 go reconfigure with override go sp_configure 'clr enabled',1 go reconfigure with override go 00 % Messages Configuration option 'clr enabled' changed from 1 to 1. Run the RECONFIGURE statement to install.	Remember that this application must migrate to Azure SQL Database Managed Instance (SQL instance as a service) as it uses CLR. CLR is NOT available on Azure SQL DB (database as a service).

Formatted: Font: 11 pt, Bold, Font color: Auto
Formatted: Font: 11 pt, Font color: Auto
Formatted: Font: 11 pt, Bold, Font color: Auto
Formatted: Font: 11 pt, Font color: Auto
Formatted: Font: 11 pt, Bold, Font color: Auto
Formatted: Font: 11 pt, Font color: Auto
Formatted: Font: 11 pt, Bold, Font color: Auto
Formatted: Font color: Accent 1

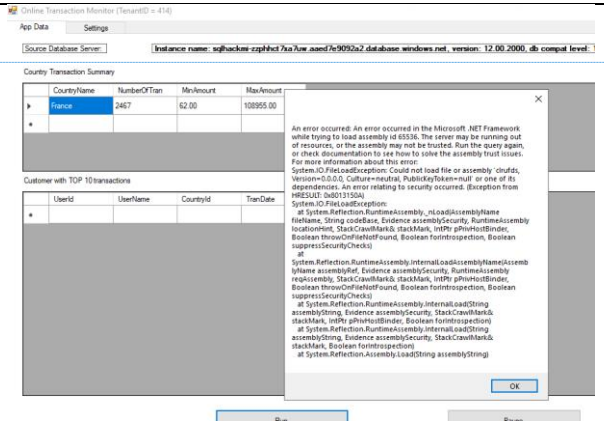
2.5. Connect 'Online Transaction Monitor' to Azure SQL DB Managed Instance

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'**		

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:** **Fully qualified Managed Instance Name:** **SqlhackmiXXXXXXX** **Initial Catalog:** **TEAM01_TenantDataDb** **UserName:** **TEAMXX** **Password:** **TEAMXX** Click '**Change Connection String**' to apply these new settings. ~~**ServerName: TargetMIhost**~~ ~~**Initial Catalog:** TenantDataDb~~ ~~**Username:** TargetMIuser~~ ~~**Password:** TargetMIpwd~~		Use the parameters from your "Workshop Sheet – Parameters".

Formatted Table
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted

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 databases don't have the correct trust settings.** **Run the 3 ALTER DATABASE statements below and try starting the application again.**	 ``` -- -- CHANGE BELOW TO YOUR TEAM NUMBER (REPLACE XX) USE [TEAMXX_TenantDataDb] GO -- EXEC dbo.sp_changedbowner 'sa' alter database [LocalMasterDataDB] set trustworthy on go alter database [SharedMasterDataDB] set trustworthy on go alter database [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.

Formatted: Font: Bold