

SQL Moderation Hack Database Migration Lab Step-by-step

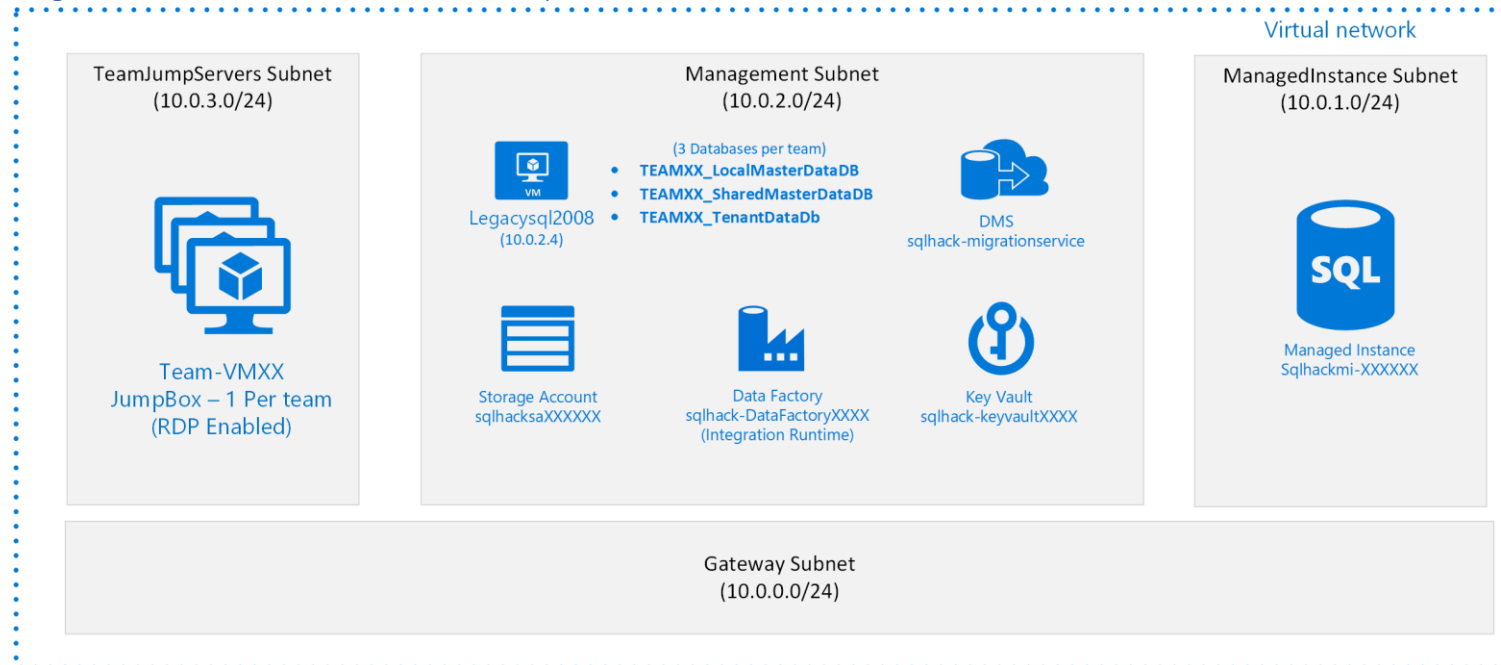
V2.4

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.....	18
4. Confirm application databases have been migrated to Azure SQL Managed Instance.....	31
5. Connect 'Online Transaction Monitor' App to Azure SQL DB Managed Instance	32



Migration architecture and Azure components



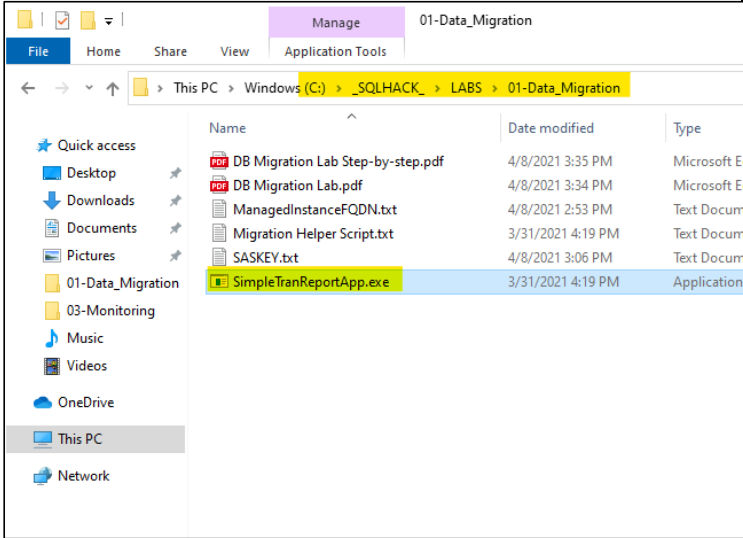
SQLHACK-SHARED-VNET		
Single Virtual Network containing all workshop resources		
TeamJumpServers Subnet Each team is assigned a Win10 VM that mimics their company desktop	Management Subnet Several machines and services are already deployed within a dedicated subnet within the Virtual Network	ManagedInstance Subnet The Azure SQL Managed Instance has been deployed into a dedicated Subnet

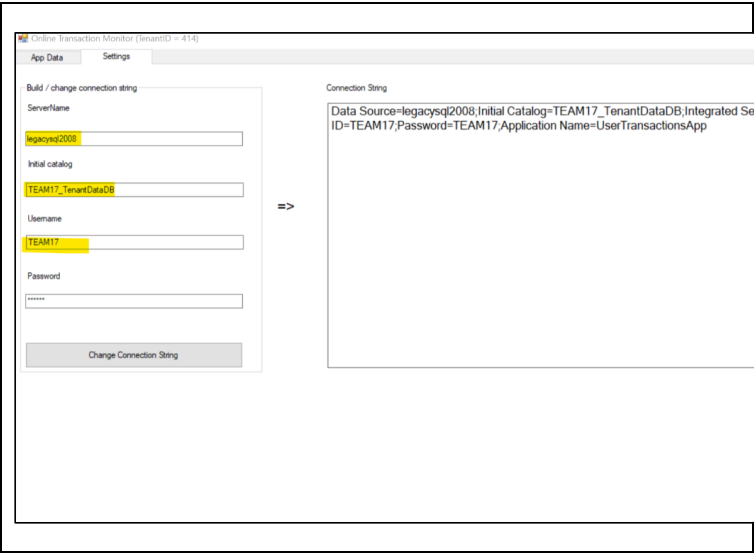
Generic Migration Content

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

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 an be found in: C:\SQLHACK_\LABS\01-Data_Migration</p> <p>Or</p> <p>In Windows Explorer search for ‘SimpleTranReportApp.exe’</p>	 <p>The screenshot shows a Windows File Explorer window titled '01-Data_Migration'. The address bar shows the path 'This PC > Windows (C:) > _SQLHACK_ > LABS > 01-Data_Migration'. The left sidebar shows 'This PC' selected. The main pane displays a list of files and folders:</p> <table border="1"><thead><tr><th>Name</th><th>Date modified</th><th>Type</th></tr></thead><tbody><tr><td>DB Migration Lab Step-by-step.pdf</td><td>4/8/2021 3:35 PM</td><td>Microsoft E</td></tr><tr><td>DB Migration Lab.pdf</td><td>4/8/2021 3:34 PM</td><td>Microsoft E</td></tr><tr><td>ManagedInstanceFQDN.txt</td><td>4/8/2021 2:53 PM</td><td>Text Docum</td></tr><tr><td>Migration Helper Script.txt</td><td>3/31/2021 4:19 PM</td><td>Text Docum</td></tr><tr><td>SASKEY.txt</td><td>4/8/2021 3:06 PM</td><td>Text Docum</td></tr><tr><td>SimpleTranReportApp.exe</td><td>3/31/2021 4:19 PM</td><td>Application</td></tr></tbody></table>	Name	Date modified	Type	DB Migration Lab Step-by-step.pdf	4/8/2021 3:35 PM	Microsoft E	DB Migration Lab.pdf	4/8/2021 3:34 PM	Microsoft E	ManagedInstanceFQDN.txt	4/8/2021 2:53 PM	Text Docum	Migration Helper Script.txt	3/31/2021 4:19 PM	Text Docum	SASKEY.txt	4/8/2021 3:06 PM	Text Docum	SimpleTranReportApp.exe	3/31/2021 4:19 PM	Application	<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>
Name	Date modified	Type																					
DB Migration Lab Step-by-step.pdf	4/8/2021 3:35 PM	Microsoft E																					
DB Migration Lab.pdf	4/8/2021 3:34 PM	Microsoft E																					
ManagedInstanceFQDN.txt	4/8/2021 2:53 PM	Text Docum																					
Migration Helper Script.txt	3/31/2021 4:19 PM	Text Docum																					
SASKEY.txt	4/8/2021 3:06 PM	Text Docum																					
SimpleTranReportApp.exe	3/31/2021 4:19 PM	Application																					

<p>Once running, select the Settings tab and enter the following parameters into the fields identified:</p> <p>ServerName: LEGACYSQL2008</p> <p>Initial Catalog: TEAMxx_TenantDataDb</p> <p>Username: TEAMxx</p> <p>Password: TEAMxx</p> <p>Click the “Change Connection String” button to apply the connection string modifications</p>		<p>Use the parameters from the Appendix in the “Hands-on Lab - Data Migration” document.</p> <p>The connection string will now have been set to connect to the legacy SQL host – LEGACYSQL2008 with appropriate Team database and login details.</p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

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

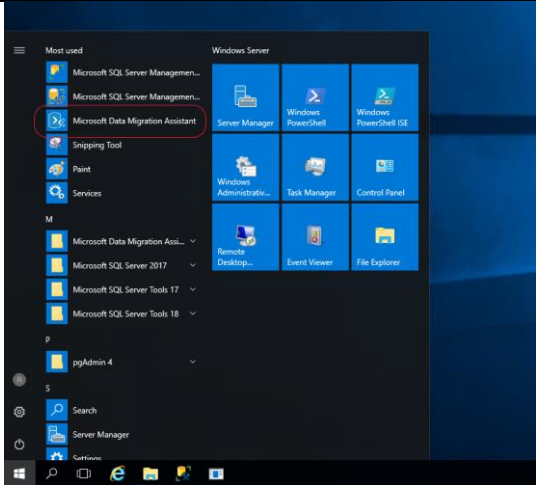
Pause

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)

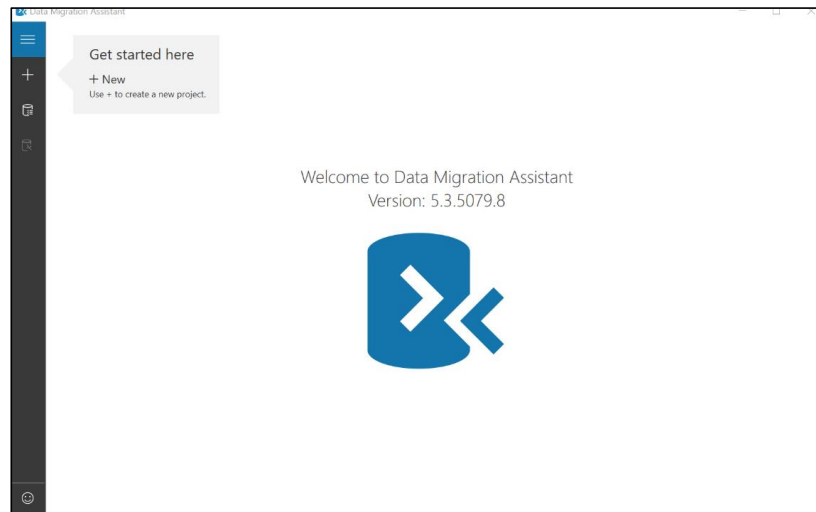
In this section we will use the Data Migration Assistant (DMA) to assess the applications database for suitability for migration to Azure Cloud.

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>	 A screenshot of a Windows 10 Start menu. The 'Most used' section is visible, showing several applications. 'Microsoft Data Migration Assistant' is highlighted with a red circle. Other applications listed include 'Microsoft SQL Server Management Studio', 'Server Manager', 'Windows PowerShell', 'Windows PowerShell ISE', 'Snipping Tool', 'Paint', 'Services', 'Task Manager', 'Control Panel', 'Remote Desktop...', 'Event Viewer', 'File Explorer', 'pgAdmin 4', 'Search', and 'Server Manager'. The taskbar at the bottom shows the Start button, Search, and several pinned applications.	<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>

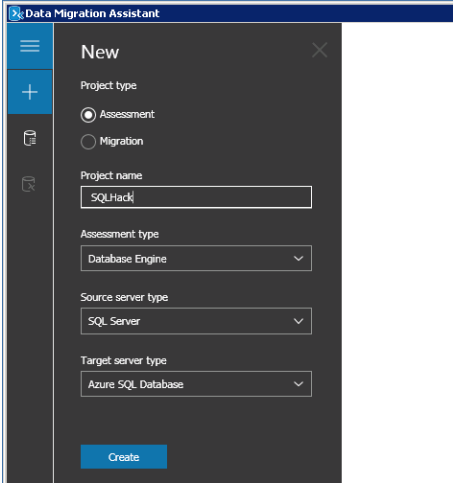
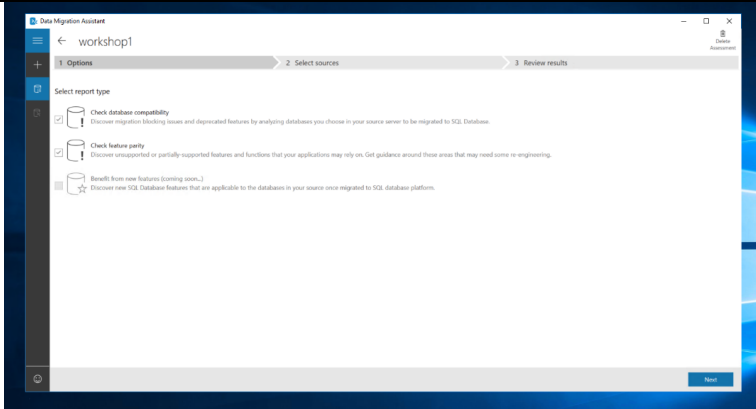
SQL Modernisation Open Hack

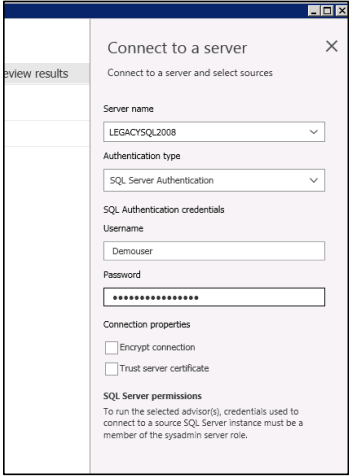
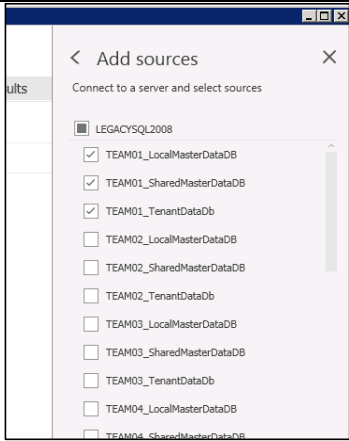
You should see this screenshot to the right.

Select the “+” to create a new **assessment** project



SQL Modernisation Open Hack

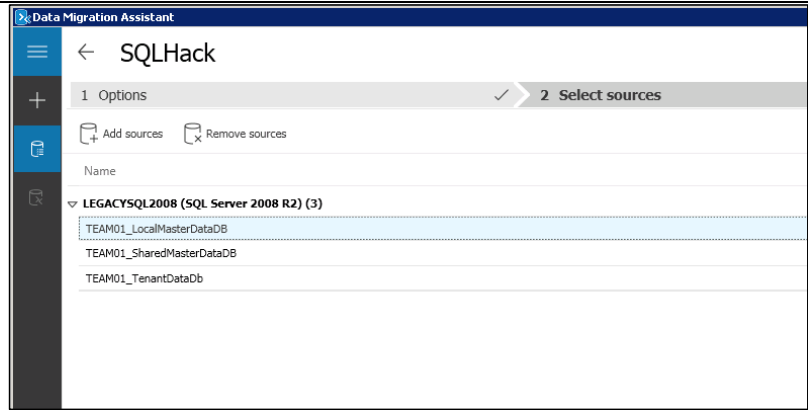
<p>Select/Enter the following details:</p> <p>Project name: Workshop1</p> <p>Assessment type: Database Engine</p> <p>Source server type: SQL Server</p> <p>Target server type: Azure SQL Database</p> <p>Click 'Create'</p>		<p>Our first project assessment assumes we will be migrating to Azure SQL DB, so the options shown in the screenshot need to be selected.</p>
<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>DMA can test for both database compatibility and feature parity compliance against the Azure target.</p> <p>As this is the initial evaluation, we are assessing a database(s) we will perform all of these 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>If you get an error logging in check that the Win10 keyboard is set to UK English not US</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 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</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>

SQL Modernisation Open Hack

You should now see the screen on the right with the relevant TEAMxx databases listed.

Select ‘Start Assessment’



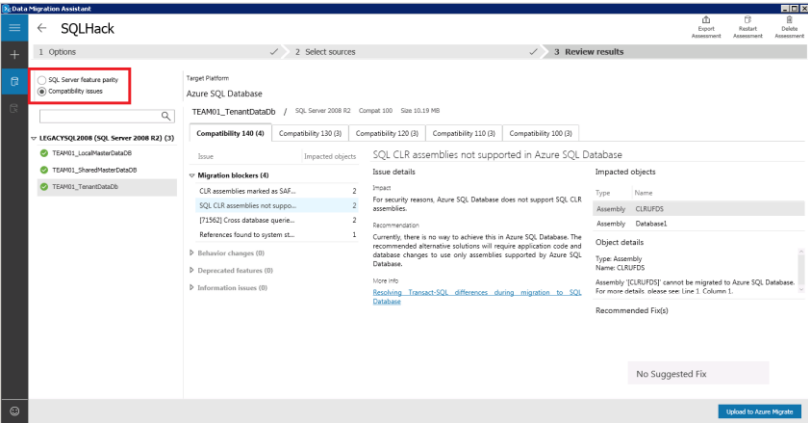
Note: DMA allows you to either ‘Add’ or ‘Remove’ additional data sources as needed at this point.

Also note that DMA provides some high-level metadata about the databases including their compatibility level the total size of each database.

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.



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

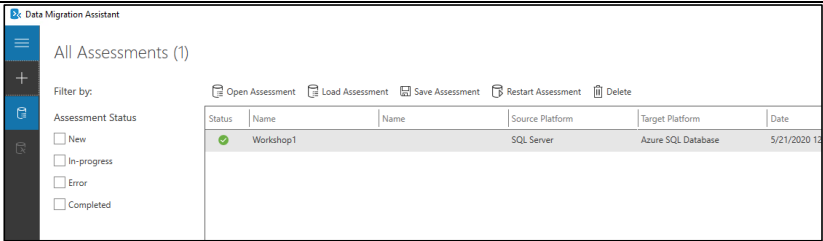
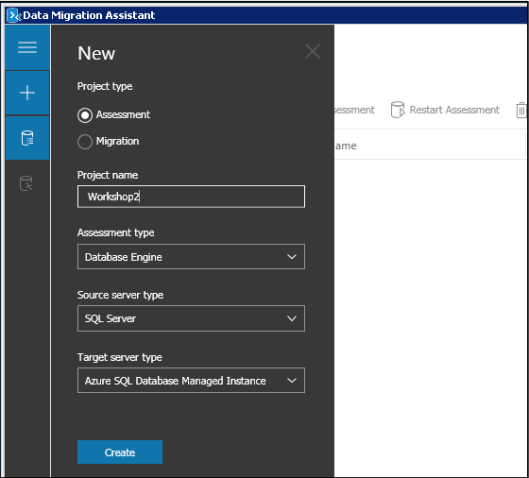
‘SQL Server feature parity’ shows what features are not supported in the target data source. Under the ‘Details’ and ‘Databases’ sections on the right you will find remedial action that are required and the databases impacted.

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

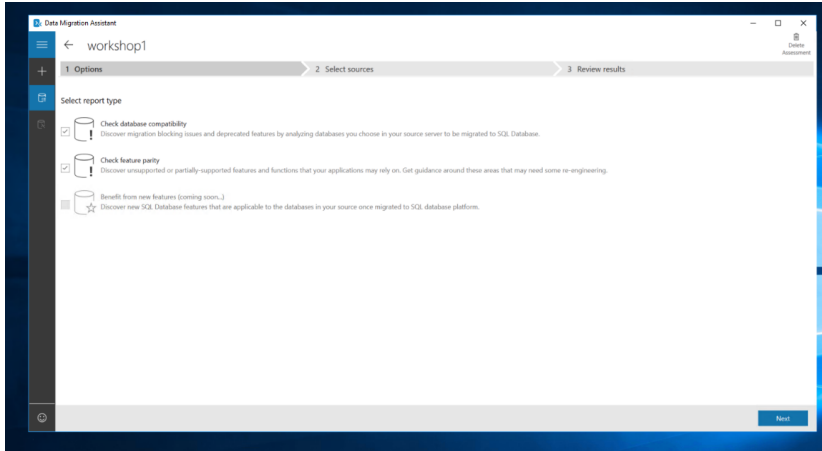
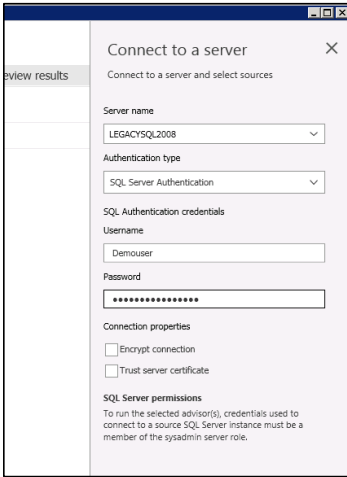
SQL Modernisation Open Hack

<p>Select 'TEAMxx_TenantDataDb'</p> <p>Note the 4 'Migration blockers' including CLR which the database uses.</p> <p>CLR is not supported on Azure SQL DB but is supported by Azure SQL Database Managed Instance (SQLMI).</p>		<p>chosen compatibility level (e.g. 140, 130, 120, 110).</p> <p>If you have multiple databases, as with the example screenshot, you need to highlight EACH database to see the compatibility issues.</p>
<p>Once you've 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>		

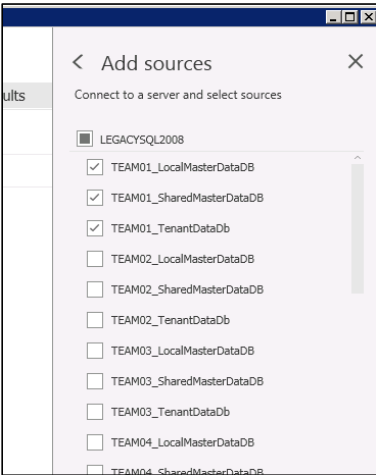
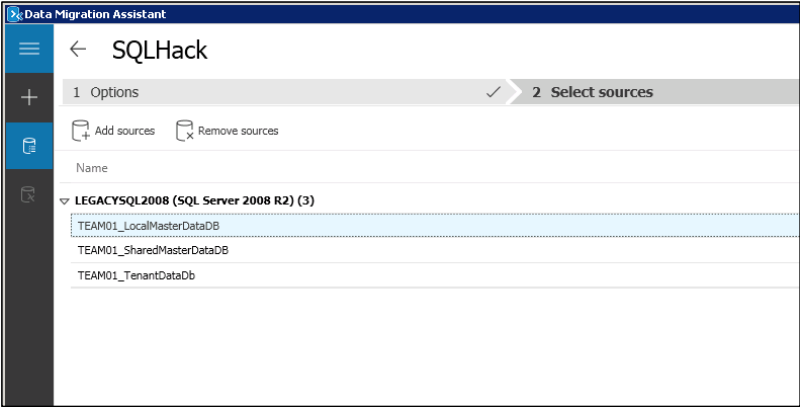
SQL Modernisation Open Hack

<p>Select the “+” to create a new assessment project.</p>		
<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>

SQL Modernisation Open Hack

<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>

SQL Modernisation Open Hack

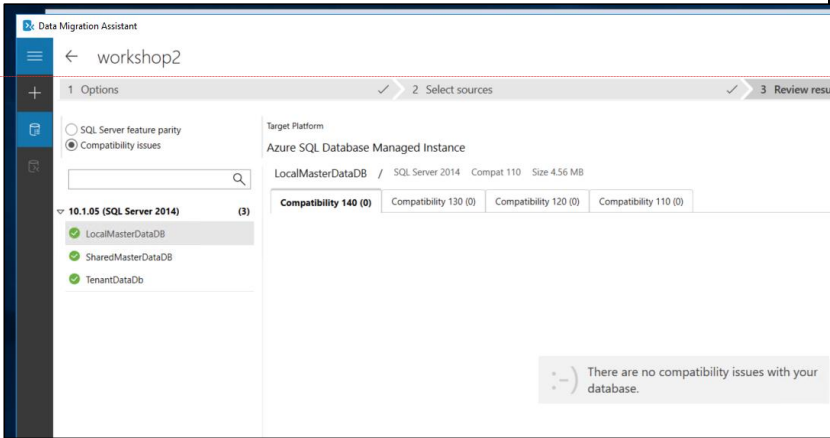
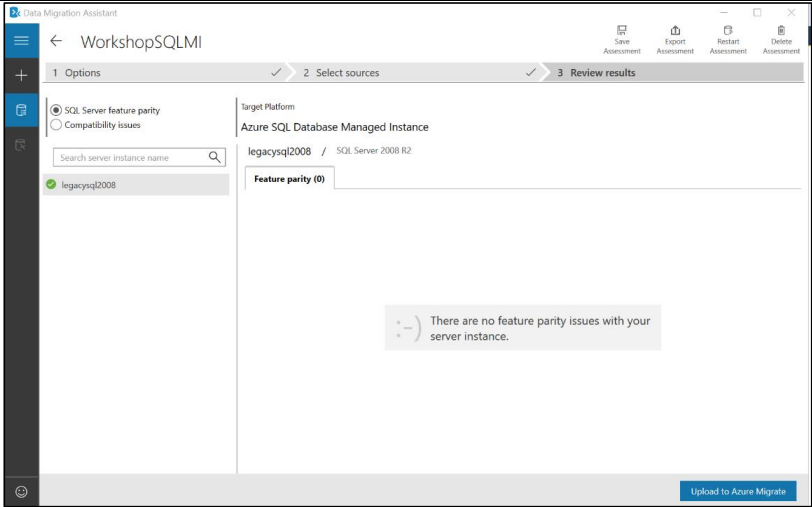
<p>Select only the 3 database used by your 'Online Transaction Monitor' app. These will have a TEAMXX prefix where XX should be replaced by your team number.</p> <p>TEAMxx_LocalMasterDataDb</p> <p>TEAMxx_SharedMasterDb</p> <p>TEAMxx_TenantDataDb</p> <p>Click 'Add' to add them to the assessment.</p>		<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>

SQL Modernisation Open Hack

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.

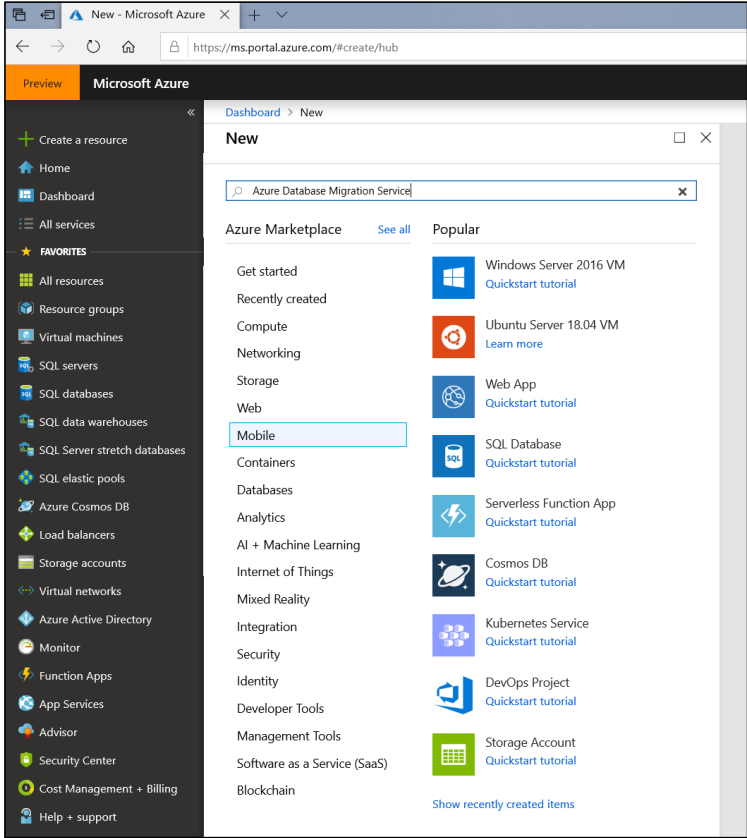
‘Compatibility Issues’ shows, over the compatibility tabs, issues that need to be addressed to permit the database(s) to run, in the chosen compatibility level (e.g. 140, 130, 120, 110).

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

Commented [SM(C1): Issue about SAFE CLR's not being migrated...

	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>

SQL Modernisation Open Hack

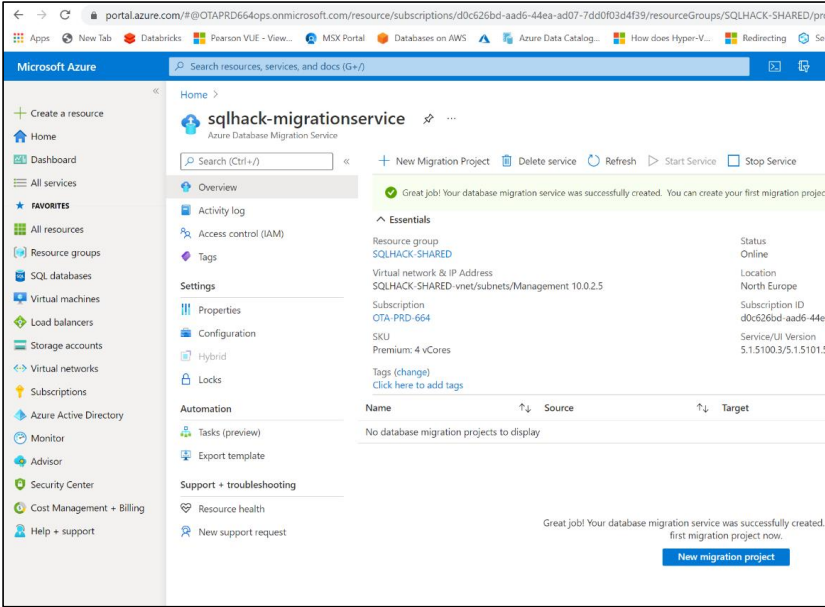
For this workshop:
On your Win10 VM open Edge browser and got to:

HTTPS://portal.azure.com

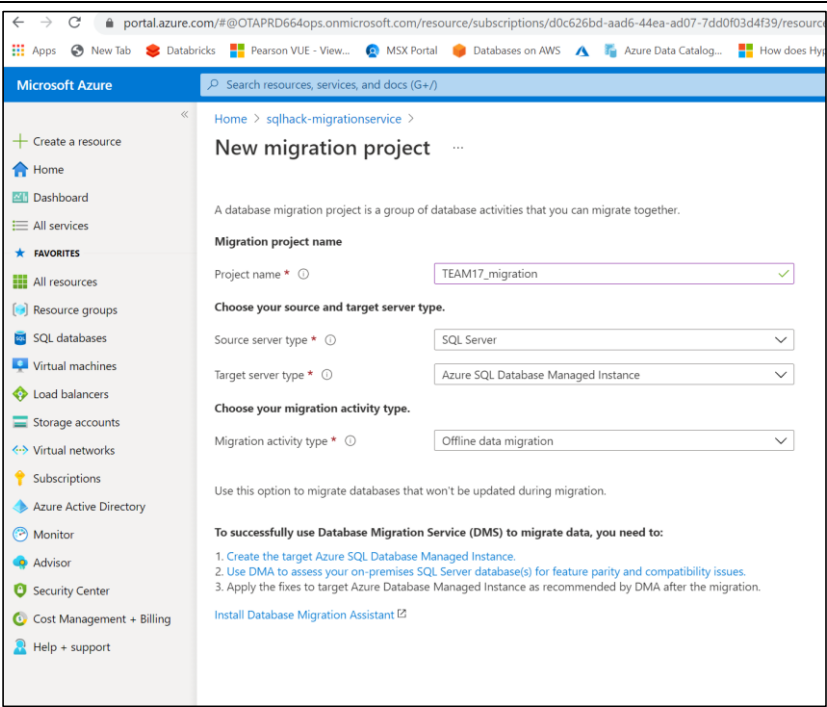
Username and Password:
(see
*C:_SQLHACK_\Azure
Portal Domain
Accounts.pdf*)

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'**



SQL Modernisation Open Hack

<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 ‘Create and run activity’</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. 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>

SQL Modernisation Open Hack

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

STEP 1: Select Source

The source is the legacy SQL VM host:

Source SQL Server:
LEGACYSQL2008

Authentication Type:
SQL Authentication

User Name:
Demouser

Password
Demo@pass1234567

Click 'Next: Select target >>'

DMS will now perform a connection test to source environment.

Microsoft Azure portal showing the 'SQL Server to Azure SQL Managed Instance Offline Migration Wizard' configuration page. The 'Select source' tab is active, showing fields for Source SQL Server instance name (legacysql2008), Authentication type (SQL Authentication), User Name (Demouser), and Password (masked). There are checkboxes for 'Encrypt connection' and 'Trust server certificate'. A message states: 'DMS requires TLS 1.2 security protocol enabled to establish an encrypted connection to the source SQL Server. Follow these steps to enable TLS support: TLS 1.2 support for Microsoft SQL Server. Or, enable TLS 1.0/1.1 from service configuration.' At the bottom are buttons for 'Review and start migration' and 'Next: Select target >>'.

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

SQL Modernisation Open Hack

STEP 2: Select Target

Target SQL Server:

(see

[C:\SQLHACK\LABS\01-Data_Migration\](#)

[ManagedInstanceFQDN.txt](#)
for the SQL Managed Instance name)

Authentication Type:

SQL Authentication

User Name:

Demouser

Password:

Demo@pass1234567

Click 'Next:Select databases>>'

DMS will now perform a connection test to target environment.

The screenshot shows the Microsoft Azure portal interface. The left sidebar contains navigation links: 'Create a resource', 'Home', 'Dashboard', 'All services', 'FAVORITES', 'All resources', 'Resource groups', 'SQL databases', 'Virtual machines', 'Load balancers', 'Storage accounts', 'Virtual networks', 'Subscriptions', 'Azure Active Directory', 'Monitor', 'Advisor', 'Security Center', 'Cost Management + Billing', and 'Help + support'. The main content area displays the 'SQL Server to Azure SQL Managed Instance Offline Migration Wizard'. The wizard has five tabs: 'Select source', 'Select target' (active), 'Select databases', 'Select logins', 'Configure migration settings', and 'Summary'. Under the 'Select target' tab, there are four fields: 'Target server name' (value: 'sqlhackmi-7js4xfhifqbi.e3982c18562e.database.windows.net'), 'Authentication type' (value: 'SQL Authentication'), 'User Name' (value: 'Demouser'), and 'Password' (value: '*****'). At the bottom of the wizard, there are three buttons: 'Review and start migration', '<< Previous', and 'Next : Select databases >>'.

SQL Modernisation Open Hack

STEP 3: Select Databases

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

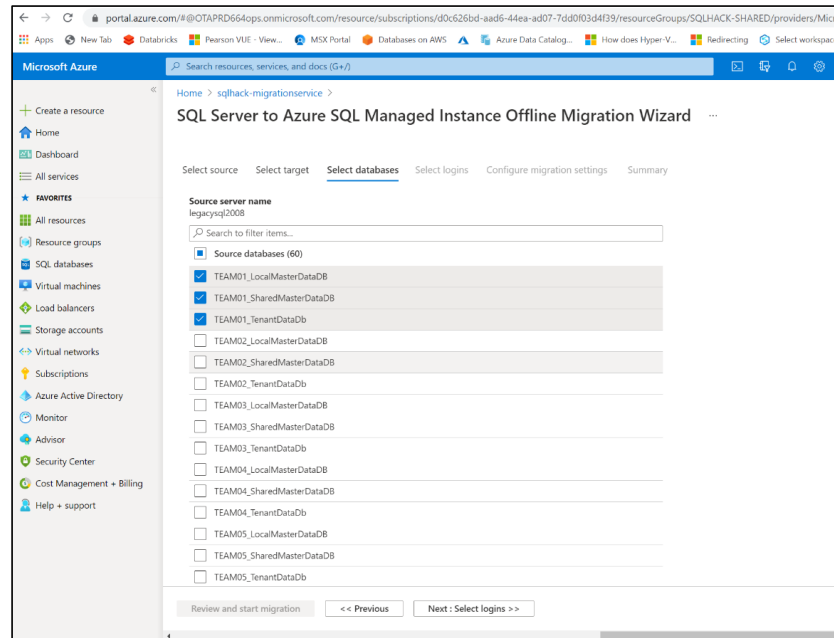
TEAMxx_LocalMasterDataDb

TEAMxx_SharedMasterDb

TEAMxx_TenantDataDb

(replace XX with your team number)

Select **'Next: Select logins>>'**



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
Next: Configure migration settings>>

Create a resource

Home

Dashboard

All services

FAVORITES

All resources

Resource groups

SQL databases

Virtual machines

Load balancers

Storage accounts

Virtual networks

Subscriptions

Azure Active Directory

Monitor

Advisor

Security Center

Cost Management + Billing

Help + support

portal.azure.com/#@CTAPR64ops.onmicrosoft.com/resource/subscriptions/dbc626bd-aad6-44ea-ad07-7dd0f03d4f39/resourceGroups/SQLHACK-SHARED/pro...

Microsoft Azure

Search resources, services, and docs (G+/)

Home > sqlhack-migrationservice >

SQL Server to Azure SQL Managed Instance Offline Migration Wizard

Select sourceSelect targetSelect databasesSelect loginsConfigure migration settingsSummary

Source server name
legacysql2008

Search to filter items...

☒

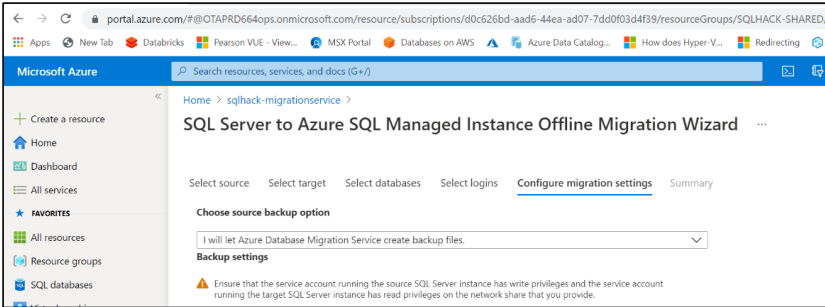
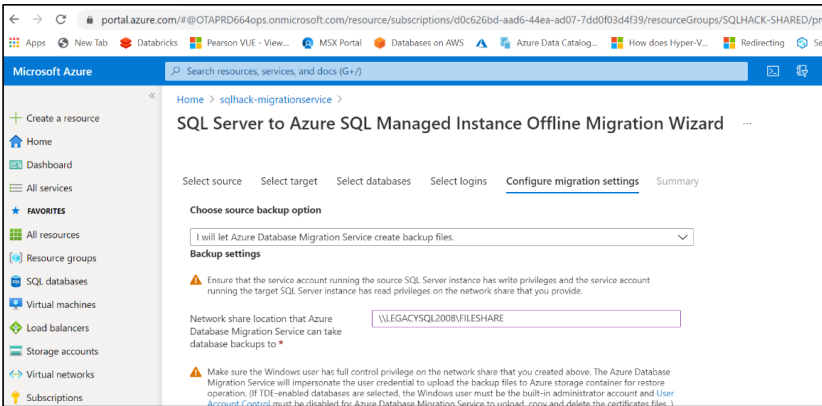
Source logins (28)

	Login type	Default database	Status
<input checked="" type="checkbox"/> TEAM13	SQL	master	Enabled
<input type="checkbox"/> TEAM09	SQL	master	Enabled
<input type="checkbox"/> TEAM05	SQL	master	Enabled
<input type="checkbox"/> NT Service\SQLIaaSE...	Windows group	master	Enabled
<input type="checkbox"/> LEGACYSQL2008\De...	Windows	master	Enabled
<input type="checkbox"/> TEAM07	SQL	master	Enabled
<input type="checkbox"/> TEAM04	SQL	master	Enabled
<input type="checkbox"/> TEAM19	SQL	master	Enabled
<input type="checkbox"/> TEAM06	SQL	master	Enabled
<input type="checkbox"/> #MS_Policy\sqlExec...	SQL	master	Disabled
<input type="checkbox"/> sa	SQL	master	Disabled
<input type="checkbox"/> TEAM15	SQL	master	Enabled
<input type="checkbox"/> TEAM10	SQL	master	Enabled
<input type="checkbox"/> TEAM03	SQL	master	Enabled
<input type="checkbox"/> TEAM11	SQL	master	Enabled

Review and start migration

<< Previous

Next : Configure migration settings >>

<p>Step 5: Configure migration Settings <i>(Choose source backup option)</i></p> <p>We are running an offline migration which will crate and use backups of the DBs that are being migrated.</p> <p>We want DMS to perform the backups, so select this option from the “Choose source backup option” (as shown).</p>		<p>DMS can use backup files provided or take the backups as part of the migration activity.</p>
<p>Step 5: Configure migration Settings <i>(Backup settings)</i></p> <p>We can now enter the Windows share that the source server will write the database backups to.</p> <p>Network Share: \\LEGACYSQL2008\FILESHARE</p>		

SQL Modernisation Open Hack

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.

Windows User Azure Database Migration Service impersonates to upload files to Azure Storage:

LEGACYSQL2008\Demouser

Password:

Demo@pass1234567

portal.azure.com/#@OTAPRD64ops.onmicrosoft.com/resource/subscriptions/d0c626bd-aad6-44ea-ad07-7dd0f03d4f39/resourceGroups/SQLHACK-SHARED/providers/Microsoft.Sqlhacking/migrationservice

Microsoft Azure Search resources, services, and docs (G+/)

Home > sqlhack-migrationservice >

SQL Server to Azure SQL Managed Instance Offline Migration Wizard

Select source Select target Select databases Select logins Configure migration settings Summary

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 *

\\LEGACYSQL2008\FILESARE

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 *

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)



Step 5: Configure migration Settings
(Storage account settings)

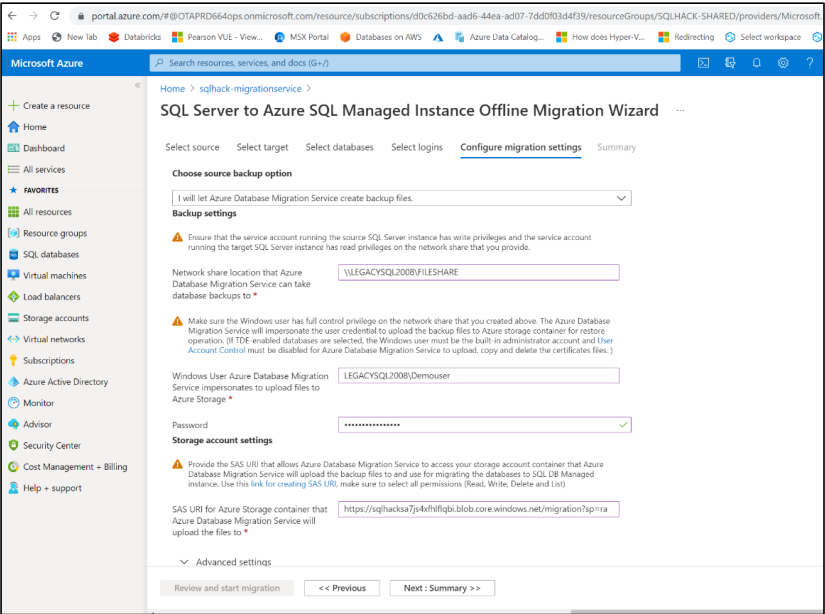
DMS is an Azure Service. We have to provide the Shared Access Signature token (or “SAS URI” for short) to permit DMS to upload the backup files from the share on the LEGACYSQL2008 host to Azure blob storage where the SQL Managed Instance can access them during the restore process.

The SAS URI is both the URL of a container (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 **‘Next : Summary>>’**.

*This will perform a connection test and if successful will display the **Summary** screen.*



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 5: Configure migration Settings (Summary)

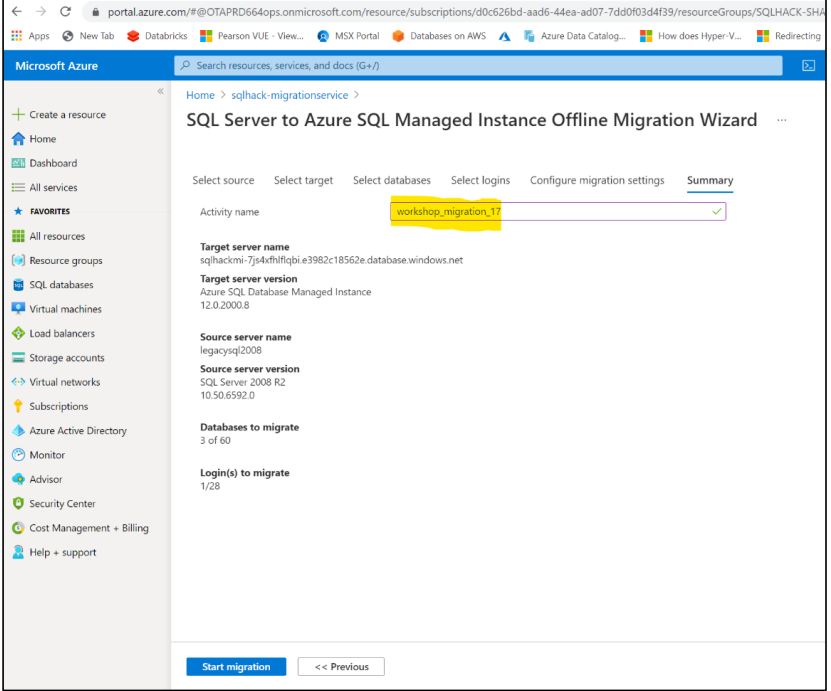
DMS displays a summary of the migration settings.

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

On the **Summary** settings enter an activity name :

Activity Name
TEAMxx_migration_activity
 (replace XX with your team number)

Click ‘**Start migration**’



The screenshot shows the 'Summary' tab of the 'SQL Server to Azure SQL Managed Instance Offline Migration Wizard'. The 'Activity name' field is highlighted with a yellow box and contains the text 'workshop_migration_17'. Other fields include 'Target server name', 'Target server version', 'Source server name', 'Source server version', 'Databases to migrate', and 'Login(s) to migrate'.

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.

SQL Modernisation Open Hack

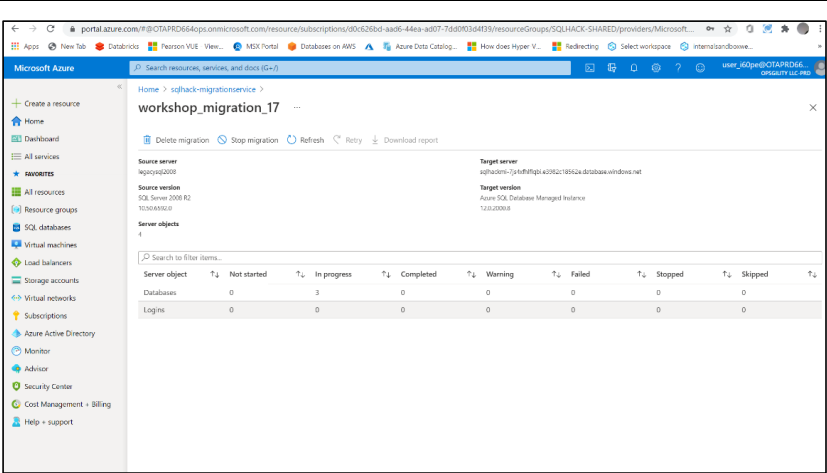
DMS will now run the migration activity.

Initially this screen will be displayed.

Click ‘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”

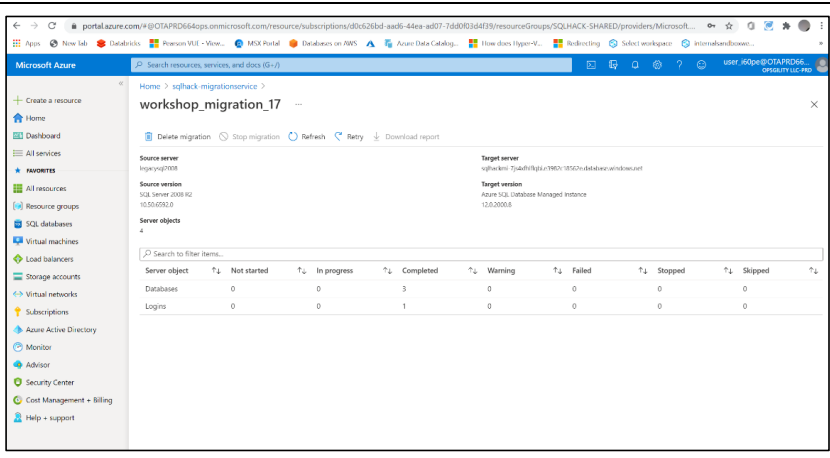


During the migration process you can monitor the creation of the 3 backup files by opening the fileshare \\LEGACYSQL2008\FILESHARE

Also note that DMS cleans-up after itself & deletes the backups from the fileshare once they have been copied up the Storage Account.

SQL Modernisation Open Hack

Under “COMPLETED”, when the number of databases says “3” the migration activity has completed.



The screenshot shows the 'workshop_migration_17' migration project in the Azure portal. The 'Server objects' table is as follows:

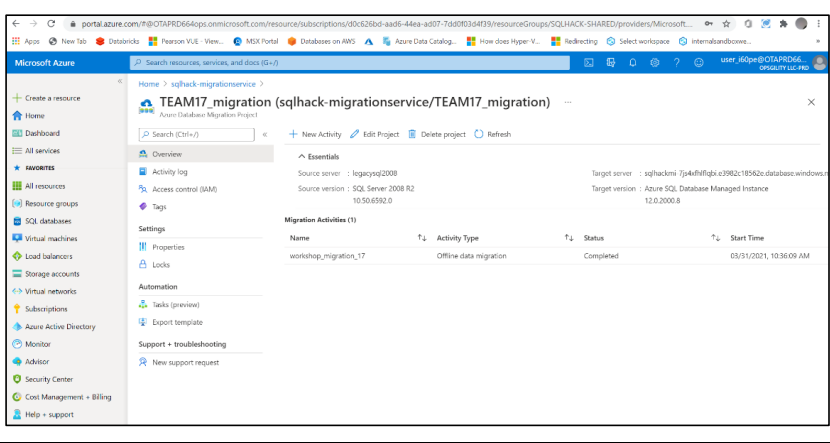
Server object	Not started	In progress	Completed	Warning	Failed	Stopped	Skipped
Databases	0	0	3	0	0	0	0
Logins	0	0	1	0	0	0	0

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 screen notice that your migration activity is displayed.



The screenshot shows the 'TEAM17_migration' project in the Azure portal. The 'Migration Activities' table is as follows:

Name	Activity Type	Status	Start Time
workshop_migration_17	Offline data migration	Completed	03/31/2021, 10:36:09 AM

DMS keeps a history of activity runs for migration projects. A migration activity can be edited and ran again.

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:

(see

C:\SQLHACK\LABS\01-Data_Migration\ManagedInstanceFQDN.txt)

SQL Authentication

Username:

Demouser

Password:

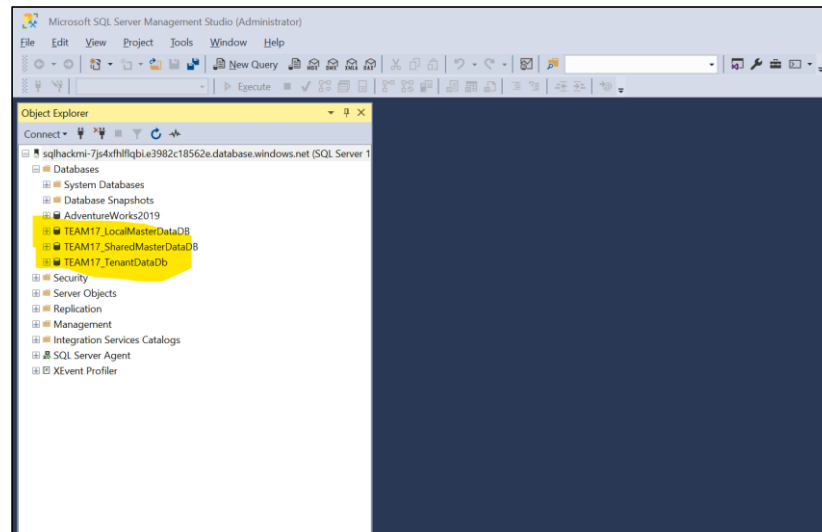
Demo@pass1234567

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

TEAMxx_LocalMasterDataDb

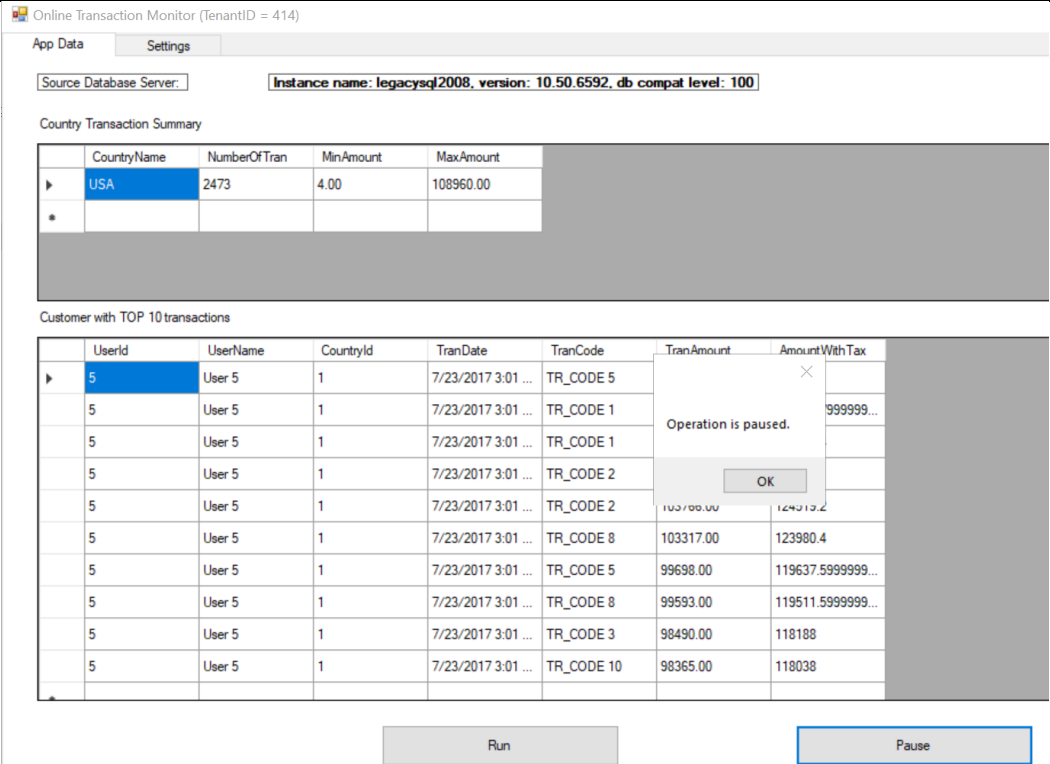
TEAMxx_SharedMasterDb

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>		

SQL Modernisation Open Hack

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:
(see [C:_SQLHACK_\LABS\01-Data_Migration\ManagedInstanceFQDN.txt](#))

Initial Catalog:
TEAMxx_TenantDataDb

UserName:
TEAMxx

Password:
TEAMxx

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

Online Transaction Monitor (tenantID = 414)

App Data

Settings

Build / change connection string

ServerName

sqlhackmi-7js4xfhflqbi.e3982c18562e.database.windows.net

Initial catalog

team17_TenantDataDB

Username

Team17

Password

Change Connection String

=>

Connection String

Data Source=sqlhackmi-7js4xfhflqbi.e3982c18562e.database.windows.net;Initial Catalog=team17_TenantDataDB;Integrated Security=False;User ID=Team17;Password=TEAM17;Application Name=UserTransactionsApp

<p>Select the 'App Data' tab Click 'Run'</p> <p>GOTCHA <i>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 in the migrated DBs.</i></p> <p>Run the 3 ALTER DATABASE statements below and try starting the application again.</p> <p>You can also find this statements already prepared in:</p> <p>C:_SQLHACK_\LABS\01-Data_Migration\Migration Helper Script.txt</p> <p>Now go back to the application and try running it again. After a few seconds you should see transactions start to appear.</p>	<div data-bbox="714 276 1326 831"></div> <pre>-- 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 EXEC sp_configure 'clr enabled', 1; RECONFIGURE; GO</pre>	<p>The application will generate simulated transactional data.</p> <p>Notice how the 'Source Database Server' connection reflects the SQL Managed Instance proving that the database migration has been completed successfully.</p>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

SQL Modernisation Open Hack

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

Online Transaction Monitor (sessionId = 414)

App DataSettings

[Source Database Server:] [Instance name: sqlm1ay-mi-055d1f61c77/database.windows.net, version: 12.00.2000, db compat level: 110]

Country Transaction Summary

CountryName	NumberOfTran	MinAmount	MaxAmount
UK	2547	16.00	102985.00

Customer with TOP 10 transactions

Userid	UserName	CountryId	TransDate	TransCode	TransAmount	AmountWithTax
1	User 8	2	7/23/2017 3:01	TRL_CODE 7	106910.00	119739.2
0	User 8	2	7/23/2017 3:01	TRL_CODE 2	106841.00	119661.92
0	User 8	2	7/23/2017 3:01	TRL_CODE 1	109904.00	119164.48
0	User 8	2	7/23/2017 3:01	TRL_CODE 3	104732.00	117299.84
0	User 8	2	7/23/2017 3:01	TRL_CODE 7	104329.00	116848.48
0	User 8	2	7/23/2017 3:01	TRL_CODE 9	103340.00	115749.76
0	User 8	2	7/23/2017 3:01	TRL_CODE 8	89112.00	111005.44
0	User 8	2	7/23/2017 3:01	TRL_CODE 3	99035.00	110919.2
0	User 8	2	7/23/2017 3:01	TRL_CODE 10	98950.00	110488
0	User 8	2	7/23/2017 3:01	TRL_CODE 2	98566.00	110393.92

RunPause